Manual do Utilizador

Commodore

1 → RECUAR A CASSETE

2 → CARREGAR O PROGRAMA
TECLAS LOAD RETURN

2 → CARREGAR NO PLAY DO
GRAGADOR

C 64

Manual do Utilizador

Commodore

Copyright © 1984 pela Commodore Business Machines, (UK) Ltd.
Reservados todos os direitos.

Este manual está protegido através de Copyright e contém informações de propriedade. Esta publicação não pode ser, na sua totalidade ou em parte, reproduzida, arquivada num sistema de repetição ou transmitida sob qualquer forma ou por quaisquer meios, electrónico, mecânico, fotocópia, gravação ou noutros termos, sem o consentimento prévio por escrito da COMMODORE BUSINESS MACHINES, (UK) Ltd.

ÍNDICE

Capítulo 1

Instalação e Operação

Desempacotar e ligar o C64 2
Instalação 3
Ligações do monitor 1084 S 5
Operação 7
Tabela para a resolução de problemas 8
Ajuste de cor 10
Ampliação do seu sistema 12

Capítulo 2

Passagem de Programas

Passagem de programas no gravador 16
Passagem de programas na disquete 17

Capítulo 3

Gravação de Programas

Guardar os programas 20
Verificação de programas gravados 21

Capítulo 4

Comandos do C64

Comandos do C64 24
Variáveis do C64 37

Capítulo 5

Mapa de Memória do Visor e das Cores

Mapa de memória do visor 40
Mapa de memória das cores 41
Pokes 43

Capítulo 6

Mensagens de "Erro"

Mensagens de erros 46

Capítulo 7

Som e música no C64

Introdução ao sintetizador do C64 50
Valores das notas musicais 53

Capítulo 8

Introdução aos Sprites

Introdução aos Sprites 58
Desenhando sprites 61
Localização dos sprites 63

Capítulo 9

Teclas com Junção T e Cores do C64

Teclar com junções 66
Utilização das cores do C64 67
Abreviaturas dos comandos basic 68

Capítulo 10

Códigos ASCII e CHR$

Códigos ASCII e CHR$ 70
Códigos do écran 73

Capítulo 11

Programas

Programas 78

AS INFORMAÇÕES NESTE MANUAL FORAM REVISTAS E SÃO COMPLETAMENTE FIDEDIGNAS. CONTUDO, NÃO SE ASSUME NENHUMA RESPONSABILIDADE POR ERROS. AS INFORMAÇÕES DESTE MANUAL ESTÃO SUJEITAS A ALTERAÇÃO SEM AVISO PRÉVIO.

ESTE MANUAL ESTÁ PROTEGIDO ATRAVÉS DE COPYRIGHT E CONTÉM INFORMAÇÕES DE PROPRIEDADE. ESTE DOCUMENTO NÃO PODE SER, NA SUA TOTALIDADE OU EM PARTE, REPRODUZIDO, ARQUIVADO NUM SISTEMA DE REPETIÇÃO OU TRANSMITIDO SOB QUALQUER FORMA OU POR QUAISQUER MEIOS, ELECTRÓNICO, MECÂNICO, FOTOCÓPIA, GRAVAÇÃO OU NOUTROS TERMOS, SEM O CONSENTIMENTO PRÉVIO POR ESCRITO DA COMMODORE BUSINESS MACHINES, INC.

Copyright © 1984 pela Commodore Business Machines (UK) Ltd.
Reservados todos os direitos.

Capítulo 1

Instalação e Operação

Instalação e Operação

- Desempacotar e Ligar o C64
- Instalação
- Ligações Opcionais
- Operação
- Tabela de Diagnósticos
- Ajustamento de Cor
- Ampliação do seu sistema com periféricos opcionais

Desempacotar e Ligar o C64

As instruções seguintes mostram-lhe como ligar o C 64 a um televisor, sistema de som, ou monitor, certificando-se de que tudo está a funcionar perfeitamente.

Antes de ligar alguma coisa ao seu computador, verifique o contéudo da embalagem do seu C64. Além deste manual, deve encontrar o seguinte:

1. Commodore C 64
2. Fonte de Alimentação (caixa cinzenta com um indicador AC e um fio de alimentação)
3. Cabo de Vídeo

Se algum destes objectos faltar deve contactar o seu dealer desde logo, para a sua substituição.

De seguida, verifique a disposição das várias ligações do computador e o que cada uma permite.

Ligações Laterais (Fig. 1)

1. **Entrada para Alimentação.** O cabo de alimentação é ligado aqui para alimentar o C64.
2. **Interruptor de Alimentação.** Liga e desliga o C64.
3. **Portas de jogos.** Cada porta de jogos pode aceitar um joystick ou um padlle, enquanto a caneta óptica só pode ser ligada à porta de jogos mais próxima da frente do C64.

Ligações Traseiras (Fig. 2)

4. **Entrada de Cartridge.** A entrada à esquerda aceita programas ou jogos em Cartridge.
5. **Selector de Canais.** Use para seleccionar qual o canal de TV que terá imagem do seu computador.
6. **Conexão de TV.** Este conector fornece imagem e som ao seu televisor.

7. **Saída de Audio e Vídeo.** Este conector fornece audio directo, que pode ser ligado a um sistema de som de alta fidelidade e um sinal de video composto que pode ser emitido para um televisor ou monitor, como por exemplo o monitor 1084 S.
8. **Porta Serie.** Pode ligar uma impressora Commodore ou uma unidade de disco (drive), directamente ao C64 através deste conector.
9. **Interface de Cassete.** Um datassete Commodore modelo 1530 pode ser ligado ao computador para se poder guardar a informação em cassete para se poder utilizar mais tarde.
10. **Porta do Utilizador.** Vários interfaces podem ser ligados a esta porta, tais como Modems ou RS 232 cartridge de comunicação.

Fig. 1

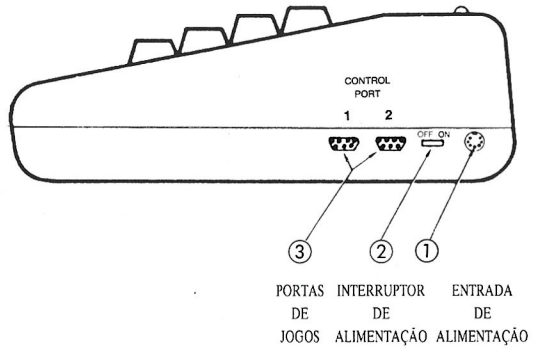
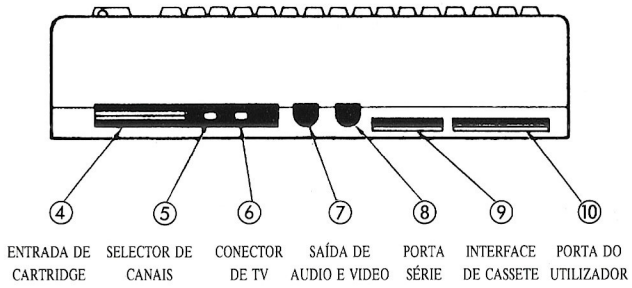


Fig. 2



Instalação

Ligações ao Televisor

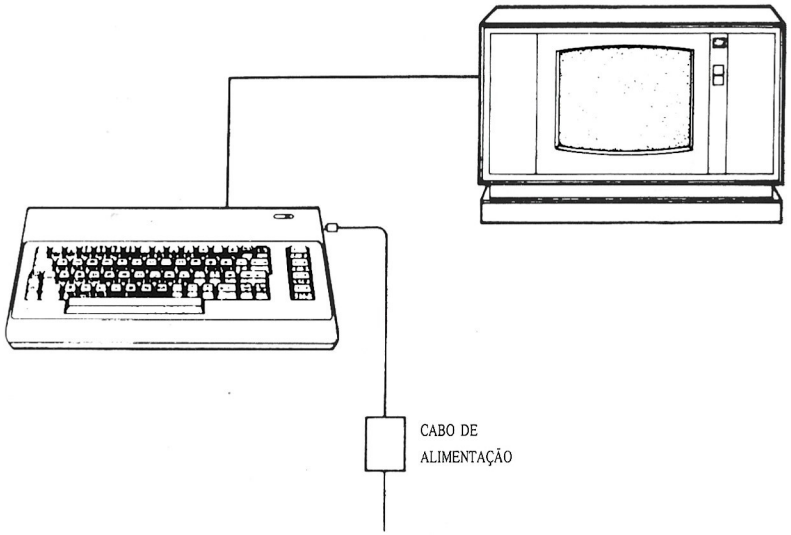
Ligue o seu computador ao seu televisor conforme se demonstra na Fig. 3.

1. Ligue uma ponta do cabo de TV à entrada de sinal TV do C64. Qualquer extremidade do cabo pode ser utilizada.
2. Ligue a outra extremidade do cabo à entrada da antena do seu televisor.

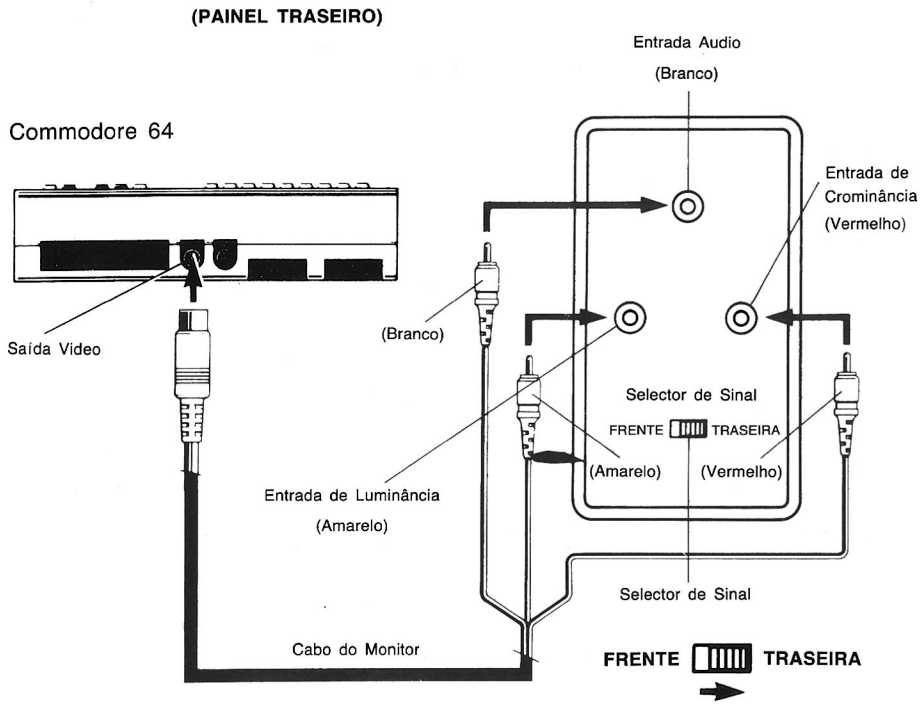
3. Coloque o cabo de alimentação na entrada de alimentação lateral do C64. Como a ficha do cabo só entra na posição certa, é impossível ligá-lo de forma errada.

O C64 está agora correctamente ligado. Não serão necessárias quaisquer ligações adicionais para usar o C64 com a TV.

Fig. 3



LIGAÇÕES DO MONITOR 1084 S



Ligações Opcionais

Como o C64 tem um canal de som de alta fidelidade, pode ligá-lo a um amplificador de som para obter a melhor fidelidade possível. Além disso, o C64 também possibilita um sinal de vídeo composto que pode ser enviado para um televisor.

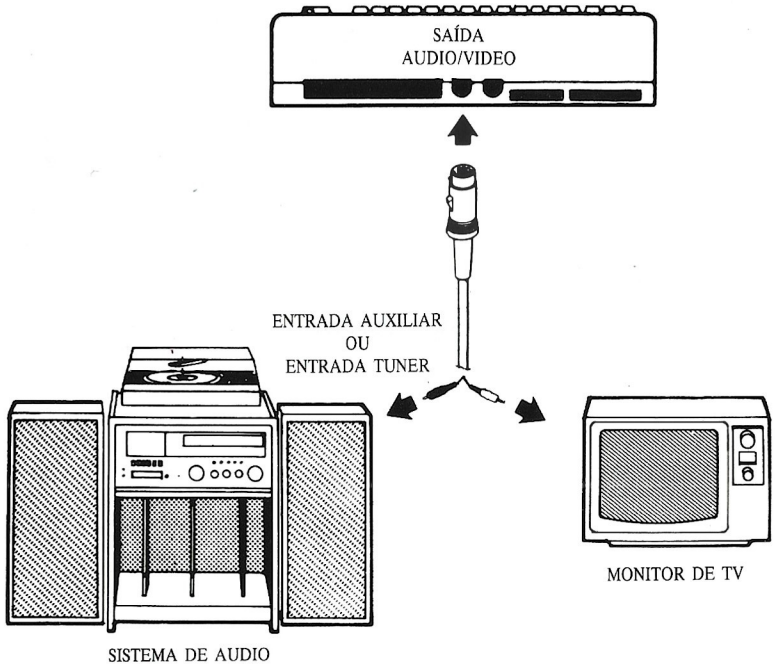
Estas opções serão possíveis através do conector audio/vídeo no painel traseiro do C64. A forma mais fácil de obter estes sinais é utilizar uma ficha DIN de 5 pinos (não fornecida) Este cabo liga directamente ao conector audio/vídeo do computador.

Normalmente o conector PRETO de ficha DIN emite o sinal AUDIO. Este conector pode ser ligado à entrada AUXILIAR de um amplificador, ou à entrada audio de um monitor ou outro sistema de vídeo tal como um gravador de video.

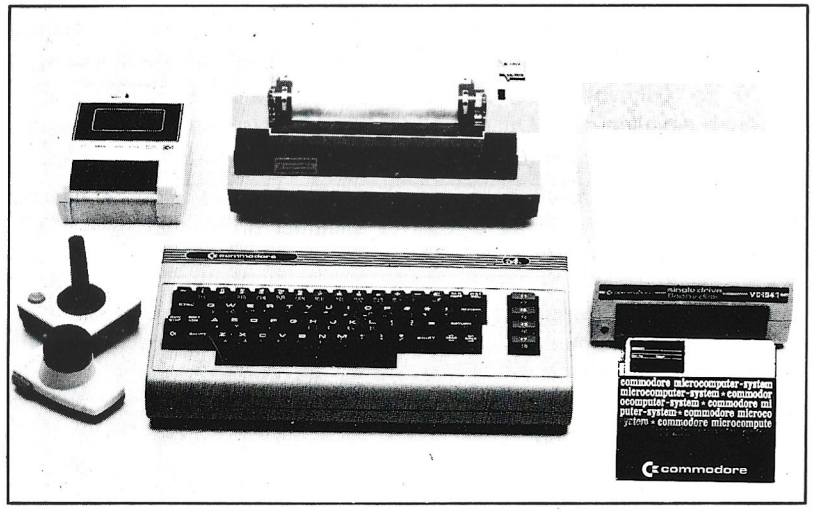
O conector BRANCO ou o ENCARNADO transporta geralmente o sinal de VIDEO directo. Este conector é ligado à entrada vídeo do monitor ou à entrada de outro sistema de vídeo tal como de um vídeo gravador.

As cores das fichas do cabo DIN podem ser diferentes, dependendo do fabricante do cabo.

Use as sugestões do Apêndice J para ligar correctamente os conectores se não obtiver sinal audio ou vídeo através das ligações sugeridas.



Para ligar outros equipamentos periféricos, tais como unidade de disco, impressoras ou monitores veja as instruções próprias de cada um desses equipamentos.



Operação

Operar com o C64

1. Ligue o computador através do interruptor do lado direito do C64.
2. Depois de alguns momentos, aparecerá no écran o que consta da Fig.5.

Fig. 5

3. Se o seu televisor tem ajuste manual de sintonia use-o para obter uma imagem nítida.
4. Pode também querer ajustar as cores e o contraste do seu televisor. Pode usar o procedimento de ajuste de cor, descrito mais adiante para tudo funcionar devidamente. Quando obtiver a primeira imagem, o écran deve aparecer **azul escuro** com bordos e letras **azuis claros.**

Se não obtiver os resultados esperados reverifique os cabos e ligações. A tabela seguinte ajuda-o a isolar e resolver qualquer problema.

TABELA PARA A RESOLUÇÃO DE PROBLEMAS

Sintoma	Causa	Solução
Luz indicadora "ON" apagada	Computador não está ligado	Verifique se o interruptor de funcionamento está em posição "ON"
	Cabo de alimentação mal ligado	Verifique entrada do cabo e religue-o
	Fonte de alimentação não funciona	Verifique a ligação à tomada da parede
	Fusível em mau estado	Leve o computador ao seu distribuidor autorizado para substituição do fusível
Ausência de Imagem	TV no canal errado	Procure outro canal que tenha imagem
	Cabo de Vídeo não ligado	Verifique a saída de TV e respectiva conexão
	Computador sintonizado para o canal errado	Sintonize o computador para o mesmo canal que a TV

Sintoma	Causa	Solução
Interferência na TV com cartridge colocados	Cartridge colocados incorrectamente	Retire o cartridge depois de desligar o computador
Imagem sem cor	TV mal sintonizada	Sintonize a TV correctamente
Imagem com cor deficiente	Ajustamentos deficientes da cor na TV	Ajuste a cor, o brilho a luminosidade através dos botões de controlo de TV
Som com excesso de ruído de fundo	Volume de TV muito alto	Ajuste o volume da TV
Boa imagem mas sem som	Volume de TV muito baixo	Ajuste o volume da TV
	Saída para o auxiliar de um amplificador incorrectamente conectada	Ligue a ficha de som à entrada auxiliar do amplificador e seleccione essa entrada auxiliar

Cursor

O quadrado intermitente por baixo da palavra READY chama-se cursor. É uma marca que mostra onde será inscrito no écran o que se digitar no teclado. Conforme se digita, o cursor move-se para a direita um espaço, consoante a sua posição original é substituída pelo carácter que se digitou. Tente digitar no teclado e observe o movimento do cursor conforme os caracteres vão aparecendo no écran.

Ajuste de Cor

Existe uma forma simples de obter uma paleta de cores no monitor para que facilmente se ajuste o sistema. Apesar de ainda não estar familiarizado com a operação do computador, atente no que se segue e verificará como é fácil o seu computador.

Primeiro, olhe para o lado esquerdo do teclado e localize a tecla marcada com CTRL . Esta tecla é chamada de CONTROL e é usada em conjunto com outras teclas para instruir o computador a fazer determinada tarefa.

Para usar a função Control, pressione a tecla CTRL , enquanto pressiona uma segunda tecla.

Experimente o seguinte: pressione a tecla CTRL enquanto pressiona a tecla 9, depois largue ambas as teclas. Nada de óbvio aconteceu, mas se carregar em qualquer tecla, o écran mostrará o carácter no modo reverso, em vez do modo normal.

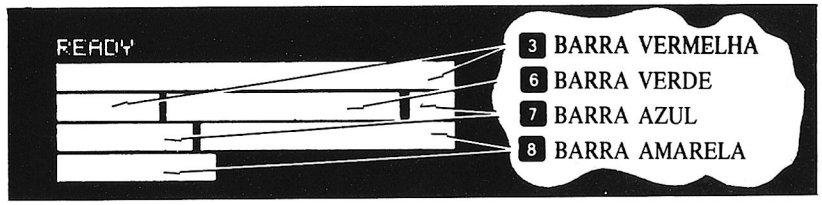
Pressione a tecla SPACE BAR . O que é que aconteceu? Se executou tudo correctamente verá uma barra azul clara mover-se ao longo do écran e para a linha de baixo enquanto a tecla SPACE BAR for pressionada.

Agora pressione CTRL enquanto pressiona qualquer das outras teclas numéricas. Cada uma tem uma cor marcada. Qualquer coisa digitada a partir daqui sê-lo-á nessa cor. Por exemplo, pressione a tecla CTRL e a tecla 8 e solte ambas. Agora pressione SPACE BAR .

Observe o écran. A barra é agora amarela, da mesma forma, pode mudar a cor para qualquer outra indicada nas teclas numéricas, pressionando CTRL e a respectiva tecla.

Mude as barras para mais algumas cores, e ajuste as cores e o brilho do seu monitor até que as cores no écran sejam as que pretende.

O écran será assim:



Nesta altura tudo está ajustado propriamente e a funcionar correctamente. Os capítulos seguintes introduzem-no na linguagem BASIC. Contudo pode imediatamente começar a utilizar algumas das aplicações pré-definidas e os jogos disponíveis sem saber algo acerca da programação do computador.

Cada um desses programas, contém informações detalhadas acerca do seu uso, contudo sugere-se que leia mais alguns capítulos deste manual para ficar mais familiarizado com o funcionamento do seu novo sistema.

Ampliação do seu Sistema com Periféricos Opcionais

A Commodore oferece uma grande variedade de periféricos que expandem as capacidades do seu computador tais como:

- Unidades de Disco (Drives)
- Impressoras e Plotters
- Monitores
- Modems para Comunicação
- Acessórios para Jogar
- Controladores de Mesa
- Moduladores Gráficos e de Som

Unidades de Disco

As unidades de disco Commodore permitem-lhe armazenar grandes quantidades de informação em disquettes de 5.25 – por exemplo a Drive 1541 da Commodore.

As disquettes permitem rápido armazenamento e consulta, e automaticamente arrumam todos os ficheiros em directórios que podem ser visionados no écran ou impressos numa impressora.

Além disso pode ligar unidades de disco em série, ligando a primeira ao computador e as restantes umas às outras.

Impressoras e Plotters

Pode ligar uma Impressora ou uma Plotter a um C64.

Monitores

O Monitor da Commodore 1084 S de 14 polegadas, oferece uma imagem colorida de alta resolução que enriquece a sua experiência com o C64. Este monitor é ligado ao computador através de um cabo DIN de 8 pinos. O manual de uso do monitor 1084 S explica claramente as ligações. Pode ainda consultar o Apêndice I para mais informações acerca do cabo DIN de 8 pinos.

Acessórios para Jogos e outros Usos

A Commodore possui joysticks e paddles que tornam mais espectaculares os jogos do Commodore C64, podendo também ser usados noutras aplicações. Podem também ser utilizadas canetas ópticas.

Conexão a um Sistema de Stereo

As capacidades de som do Commodore C64 podem ser melhoradas ligando o seu computador a um amplificador de alta fidelidade bem como a colunas de som estereofónico. Um cabo DIN de 8 pinos pode ser usado para ligar o seu computador a um amplificador.

A Commodore oferece uma variedade tão grande de periféricos que lhe permite criar o seu próprio sistema computorizado. Para mais informações acerca dos periféricos Commodore leia o guia de periféricos Commodore, e as revistas Commodore referidas no Apêndice X e contacte o seu distribuidor Autorizado Commodore.

Capítulo 2

Passagem de Programas

Passagem de Programas no Gravador

Colocar a cassete no gravador e verificar se está toda bobinada para o lado esquerdo. Caso contrário, utilize a tecla de retrocesso no gravador ←.

A operação que se vai seguir é usualmente chamada de "carregar programas", isto é, passar um programa que se encontra gravado numa cassete para a memória do computador. Isto pode ser feito de duas formas:

1. Escreve-se o comando LOAD, seguido da tecla RETURN;
2. Pressiona-se a tecla SHIFT + RUN/STOP e surgirá no écran a seguinte mensagem:

```
LOAD
PRESS PLAY ON TAPE
```

O Computador fica à espera que pressione a tecla PLAY no gravador, de modo a que a cassete comece a rodar.

Aguarde até que surja a mensagem:

```
SEARCHING
FOUND PROGRAMA
```

O aparecimento desta mensagem indica que o programa foi encontrado e que vai começar a sua passagem para a memória do Computador.

Se não quiser esperar que o gravador retome a marcha, passados alguns segundos, então pressione a tecla com o símbolo C= e o programa começará a passar.

ATENÇÃO:

Não tocar no Computador enquanto este carrega os programas, pois poderá destruir o acesso à cassete, o que significa que não poderá mais utilizar tal programa.

Aguarde até o gravador disparar. Isto significa que todo o conteúdo da cassete foi transferido para a memória do computador.

Passagem de Programas da Disquete

Introduzir a disquete na unidade e, para passar o primeiro programa que seja encontrado, escrever o seguinte comando:

```
LOAD "*", 8
```

Para carregar um programa específico na memória do computador, escrever o nome com que foi gravado com o comando SAVE, entre aspas "...", ou seja, por exemplo: "teste", 8.

Para fazer a listagem dos programas gravados na disquete, faça:

```
LOAD "$", 8 premindo seguidamente RETURN
```

Surgirá no écran:

```
SEARCHING FOR $
LOADING
READY
```

Uma vez visualizado este écran, editar o comando LIST, seguido de RETURN e aparecerá no display a listagem de todos os programas que lá se encontram, o que é vulgarmente conhecido por directório.

Capítulo 3

Gravação de Programas

Guardar os Programas

Se o computador não tivesse a hipótese de gravar os programas que fazemos, certamente de nada valeria a sua aquisição, pois teríamos que passar os programas todas as vezes que o desligássemos.

O *COMMODORE 64* tem duas formas de guardar os programas residentes em sua memória, utilizando para o efeito o **Gravador** ou a **Disquete**.

Para gravar o pequeno programa que se segue, teríamos que respeitar os passos abaixo indicados:

1. Bobinar a fita da cassete que se encontra no gravador, para o início, com a tecla de retrocesso;
2. Escrever o seguinte comando:

   SAVE, seguido de RETURN, surgindo no écran:

   SAVE
   PRESS RECORD AND PLAY ON TAPE

Então teríamos que pressionar a tecla de RECORD + PLAY, até o programa ser gravado.

PROGRAMA DE DEMONSTRAÇÃO

```
10 PRINT "SHIFT + CLR/HOME": REM LIMPA ECRAN
20 FOR I=1024 TO 1063:POKE I,102:GOSUB 100:NEXT I
30 FOR I=1063 TO 2023 STEP 40:POKE I,102:GOSUB 100:NEXT I
40 FOR I=2022 TO 1984 STEP-1:POKE I,102:GOSUB 100:NEXT I
50 FOR I=1944 TO 1064 STEP-40:POKE I,102:GOSUB 100:NEXT I
60 GET A$:IF A$="" THEN 60
100 FOR T=1 TO 150:NEXT T
110 RETURN
```

Para gravar o programa para a disquete, o comando seria:

SAVE "demonstração", 8

Surgindo em écran a mensagem:

OK
READY

Verificação de Programas Gravados

Depois de termos o programa gravado, podemos fazer a sua verificação para termos a certeza da qualidade da gravação. Proceda do seguinte modo:

1. Bobinar a cassete para o inicio, utilizando a tecla ← ;
2. Escrever o comando VERIFY, seguido de RETURN, pressionando a tecla PLAY no gravador, surgindo no écran:

   VERIFY
   PRESS PLAY ON TAPE
   OK
   READY

Caso não surja esta mensagem, algo de errado se passou na gravação, o que significa que teremos de voltar a gravar o programa com o comando SAVE "demonstração", – RETURN

Caso a verificação seja na disquete, far-se-ia:

VERIFY "demonstração",8, RETURN

Visualisará em écran:

VERIFY "demonstração",8
OK
READY

Capítulo 4

Comandos do "Commodore 64"

Comandos do *COMMODORE 64*

CONT (Continuar)

Este comando é usado para reiniciar o programa onde este foi interrompido com a tecla STOP ou END.

CONT não trabalha caso o programa tenha um erro.

A mensagem enviada para o écran, caso seja detectado um erro, é a seguinte:

CAN/T CONTINUE ERROR

LIST (Listar)

Este comando é usado de várias formas para fazer listagens no écran ou impressora.

LIST	Faz a listagem completa do programa
LIST X	Lista a linha número X
LIST–X	Lista até à linha X
LIST X–	Lista a partir da linha X
LIST X-Y	Lista da linha X até à Y

LOAD (Carregar)

Este comando é usado para transferir programas da DISQUETE ou GRAVADOR para a memória do computador.

LOAD procura no gravador o primeiro programa e carrega-o em memória.

LOAD "teste" procurará o programa chamado teste e carrega-o.

LOAD "teste",8 procurará o programa em DISQUETE e carrega-o.

LOAD "*",8 procura o primeiro programa que encontrar na DISQUETE e carrega-o para a memória.

RUN (Correr)

Este comando executa um programa residente na memória do computador.

RUN 100 Inicia o programa na linha 100

Uma má utilização deste comando dá o seguinte erro:

UNDEFINED STATEMENT ERROR

SAVE (Reter)

Este comando guarda programas residentes na memória do computador na DISQUETE ou no GRAVADOR.

SAVE	Guarda um programa no gravador sem o identificar por nome.
SAVE"teste"	Guarda um programa no gravador chamado teste.
SAVE"teste",8	Guarda um programa na disquete chamado teste.

VERIFY (Verificar)

Este comando faz a verificação dos programas guardados no gravador ou na disquete.

VERIFY	Procura o programa no gravador e faz a verificação da gravação.
VERIFY "teste",8	Verifica na disquete a gravação do programa chamado teste.

GET (Obter)

Este comando espera um carácter vindo do teclado.

```
10 PRINT "PRESSIONE UMA TECLA"
20 GET A$:IF A$= "" THEN 20
30 PRINT A$;
40 GOTO 20
```

GET # 1

Este comando é igual ao anterior, só que permite fazer a leitura de um carácter da file.

GOTO (Ir para)

Este comando inicia um programa sem limpar as variáveis carregadas do programa.

GOTO X Inicia o programa na linha X

GOSUB (Ir para a Subrotina)

Este comando é parecido com GOTO só que ao ser executado, o computador lembra-se da última linha donde saiu para, mediante um comando RETURN, regressar à linha seguinte.

GOSUB X O programa salta para a subrotina X

POKE (Forçar)

Esta instrução serve para carregar um valor na memória, cujo endereço seja especificado.

ATENÇÃO

A má utilização deste comando pode destruir dados ou partes do programa.
A sua sintaxe é a seguinte:

POKE endereço, conteúdo

POKE 1504,81 desenha um círculo no visor

CLOSE (Fechar)

Este comando fecha todos os canais que tenham sido abertos pela instrução OPEN

O número que se seguir ao comando CLOSE tem de ser o mesmo que foi usado para abrir o canal.

CLOSE 1 Fecha o canal número 1
Se utilizou:
OPEN 15,8,15
Fará:
CLOSE 15, fechando assim o canal da disquete.

CLR (Limpar)

Este comando limpa todas as variáveis da memória do computador, mas não destrói o programa.

Exemplo:

```
10 CLR
```

CMD

Este comando envia todo o programa para uma saída, que poderá ser uma impressora ou écran.

```
OPEN 1,4
CMD 1
LIST
```

O programa é listado na impressora.

END (Fim)

Quando este comando é encontrado o programa pára.
O programa pode ser reiniciado com o comando CONT.

INPUT (Entrada)

Este comando é usado para dar entrada de dados ou valores numéricos.
Quando o programa encontra este comando, pára e surgirá na écran o sinal ?
esperando uma entrada vinda do teclado.

Exemplo:

```
10 INPUT "A sua idade por favor";A$
20 PRINT A$
```

INPUT 1,A

Esta instrução é igual à anterior, só que em vez de esperar uma entrada do teclado, faz a leitura duma file previamente aberta com o comando OPEN.

Exemplo:

```
OPEN 1,8,1
INPUT #  1,A$
```

ON GOTO

ON GOSUB

Estes comandos são utilizados com as instruções GOTO e/ou GOSUB.

A sua utilidade está em conseguirmos enviar para uma linha de utilização do programa, uma entrada vinda do teclado.

```
10 INPUT X
20 ON X GOSUB 100,200
```

ou

```
10 INPUT X
20 ON X GOTO 100,200
```

OPEN (Abrir)

Este comando do *COMMODORE 64* serve para abrir os canais de serviço para a impressora, gravador, disquete ou écran.

ATENÇÃO

Depois de sairmos de um programa onde tenha sido utilizado este comando, **não esquecer de fechar o canal com a instrução CLOSE**.

OPEN 1,0	Abre o canal do visor
OPEN 2,1,0,"X"	Abre o canal do gravador para leitura
OPEN 2,1,1,"X"	Abre o ficheiro no gravador para escrita
OPEN 3,4	Abre o canal para a impressora
OPEN 4,8,15	Abre o canal para a disquete

RETURN (Retorno)

Este comando é utilizado em conjunto com a instrução GOSUB.

Quando num programa aparece este comando, o computador regressa à linha seguinte donde saiu com o comando GOSUB.

STOP (Parar)

Este comando pára a execução de um programa, dando a seguinte mensagem em écran:

BREAK IN XX

O Programa pode ser recomeçado com a instrução CONT.

DEFFN (Definir função)

A utilização desta função é definir uma fórmula complexa que constará do programa.

Exemplo:

```
DEF FN Y(X)=2*(10.25+X/0.5)
PRINT FN Y(7)
```

DIM (Dimensionar)

Esta é a única instrução de declaração da linguagem BASIC.

Serve para definir um número máximo de elementos de uma tabela.

A utilização deste comando reserva no interpretador lugar para armazenar os diferentes elementos deste conjunto a tratar.

A sua sintaxe é a seguinte:

```
DIM A$ (10,10)
DIM Y(X)
```

LET (Seja)

Este comando pode ser omitido no seu computador *COMMODORE 64*

```
10 LET A$="ROTINA"
20 LET B$="PORTO"
30 LET X=159
40 LET Y=45
50 PRINT A$,B$,X+Y
```

Para obter tal resultado, bastaria escrever:

```
10 A$="ROTINA"
20 B$="PORTO"
30 X=159
40 Y=45
50 PRINT A$,B$,Y+X
```

DATA (Dados)

Este comando é seguido de uma lista de valores que podem ser numéricos ou alfanuméricos.

Esta instrução é utilizada com o comando READ

Exemplo:

```
10 FOR I= 1 TO 5
20 READ X$(I)
25 PRINT X$(I)
30 NEXT I
40 DATA 10,20,30,40,50
50 END
```

PRINT (Imprimir)

Este comando é dos mais úteis em programação, pois, pode ser utilizado em conjunto com várias funções, textos, etc.

Exemplo:

```
10 INPUT"IDADE";id
20 PRINT "A sua idade é";id
30 INPUT "O seu nome é",A$
40 PRINT "Chamas-te",A$
```

PRINT # 1,A$

Poucas diferenças existem entre este comando e o anterior.

A diferença é que em vez de imprimir no écran, imprime valores para uma determinada file.

Este comando trabalha em conjunto com OPEN

```
10 OPEN 1,2,1,A$
20 PRINT # 1,A$
30 CLOSE 1
```

ATENÇÃO

Este comando não trabalha com TAB, POS SPC.

REM (Remarca)

Este comando quando utilizado num programa tem carácter meramente informativo, o que permitirá colocar esclarecimentos no programa onde vir que estes se mostram necessários. O programa não lê esta instrução.

READ (Ler)

Este comando é usado para ler o conteúdo das instruções DATA.

RESTORE (Repôr)

Este comando quando é executado, repõe o cursor na linha de leitura do comando DATA

ASC (Código ASCII)

O resultado desta função dá um número decimal que representa o código ASCII

```
10 PRINT "pressione uma tecla"
20 GET A$:IF A$=      THEN 20
30 Y=ASC(A$)
40 PRINT "O código ASCII de";A$ "E";Y
50 GOTO 10
```

CHR$(X)

Esta função faz conversão decimal para o código ASCII

```
10 PRINT CHR$ (147): REM limpa écran
20 PRINT CHR$ (65): REM imprime a letra A
```

Existem no *COMMODORE 64* três funções que permitem extrair caracteres a partir de uma cadeia existente no programa. São elas:

LEFT $ (X$,Y)
RIGHT $ (X$,Y)
MID $ (X$,X,Y)

LEFT$ (X$,Y) (Esquerda)

Esta função é parecida com a que se segue, pois permite extrair X caracteres mais à esquerda.

```
10 A$="ROSA MARIA"
20 B$="PORTUGAL"
30 PRINT LEFT$(A$,4)
40 PRINT LEFT$(B$,3)
```

RIGHT$ (X$,Y) (Direita)

Esta função permite extrair Y caracteres mais à direita da cadeia X$.

```
10 A$="LISBOA"
20 B$="PISTA"
30 PRINT RIGHT$(A$,3)
40 I$=RIGHT$ (A$,3)
50 Y$=B$+I$
50 PRINT Y$
```

MID$ (X$,X,Y)

Esta função permite extrair uma cadeia de caracteres dentro de outra.
X$ designa a cadeia de caracteres
X é um inteiro que indica a partir de que carácter deve começar a extracção.
Y indica o número de caracteres a extrair

Exemplo:

```
10 X$="LISBOA"
20 PRINT MID$ (X$,4,3)
30 END
```

SYS (Sistema)

Este comando é seguido de um número decimal que pode ir de 0 a 65535.
O programa então executa-se em linguagem máquina.

LEN(X$) (Comprimento)

Esta função permite ler o número de caracteres contidos numa cadeia.

```
10 A$="PORTUGAL"
20 PRINT LEN (A$)
```

WAIT (Espere)

Esta instrução permite obter uma pausa ou atraso previamente definido

USR(X)

Esta instrução permite um salto para um subprograma
O parâmetro X é passado para o programa em linguagem máquina.

IF..(se) THEN..(então)

É graças a estas instruções que nos é possível definir certos dados ou variáveis, pois permite-nos verificar a condição num programa.

FOR..(para) NEXT..(próximo)

Esta instrução permite programar um determinado número de ciclos.
Este comando tem necessariamente de terminar com a função NEXT

Exemplo:

```
10 PRINT CHR$(147)
20 FOR I=1 TO 20
30 PRINT 1+2
40 FOR PAUSA=1 TO 500:NEXT PAUSA
50 NEXT I
```

FRE(X) (Livre)

Esta função verifica o número de bytes disponíveis na memória do computador.

POS(x) (Posição)

Esta função regressa ao número da coluna entre (0 e 39) que o comando PRINT imprime no écran.

O valor de x não é usado no programa.

```
PRINT POS(23)
```

SPC(n) (Espaço)

Esta função imprime n espaços

```
PRINT SPC(19)
```

TAB(X) (Tabulação)

Esta função também se usa com o comando PRINT e permite imprimir na coluna x

STR$(X) (String)

Esta função converte um valor numérico em string

```
X$=STR$(x)
```

VAL(x$) (Valor)

Esta função converte um valor de string em número.
VAL(X$) é o inverso de STR$(X)

PEEK(x) (Tomar)

Esta função permite a leitura de certas posições de memória.

O valor de x quando lido é convertido em decimal e guardado numa varíavel.

Este comando é usado com a instrução POKE

RND(X) (Aleatório)

Esta instrução dá um número ao acaso que pode ser 0 ou 1.

As vantagens deste comando é poder criar uma sequência de números aleatórios.

No totoloto os números variam de 1 a 49 o que representaria o seguinte programa:

```
10 FOR I= 1 TO 6
20 X=INT((RND(1)*49)+1)
30 PRINT X;"";
40 NEXT I
50 END
```

ATENÇÃO

Neste programa não excluímos a hipótese de sairem números iguais.

ABS(x) (Valor absoluto)

Esta função regressa para o valor absoluto de um número.
O resultado é sempre positivo.

ATN(x) (Arco tangente)

Regressa a um ângulo medido em radianos, cuja tangente é x.

SGN(x) (Sinal)

Esta função regressa a um sinal POSITIVO, NEGATIVO ou ZERO de x.
O resultado é positivo se +1, 0 se zero ou −1 se negativo.

SQR(x) (Raiz quadrada)

Esta função dá raiz quadrada de x onde x é um número positivo ou zero.
Se x fôr um valor negativo, o computador indica erro.

COS(x) (Seno)

Esta função pode ser o seno de x, onde x é um ângulo em radianos.

COS(x) (Coseno)

Regressa ao valor de x, onde é x o ângulo medido em radianos.

EXP(x)

Função exponencial

LOG(x)

Logaritmo de x

TAN(x)

Tangente de x

INT(x)

Dá um valor inteiro de x

FN XX(Y)

Permite utilizar uma função previamente definida com DEF FN XX(Y)
FN XX é a função definida pelo programador. Y é uma variável e pode ser substituída por um número inteiro.

Variáveis do *COMMODORE 64*

O BASIC do *COMMODORE 64*, usa três tipos de variáveis, que são:

Variáveis Reais: Pode ser uma letra de A a Z, seguida de um número de 0 a 9.

Variáveis Inteiras: A%,. X1%, Y2%

Variáveis de String: A$,B$,A1$

ATENÇÃO

Há três variáveis que são reservadas para uso do *COMMODORE 64* e não podem ser definidas.

São elas:

ST, TI, TI$

Operadores Lógicos

Os três operadores lógicos são:

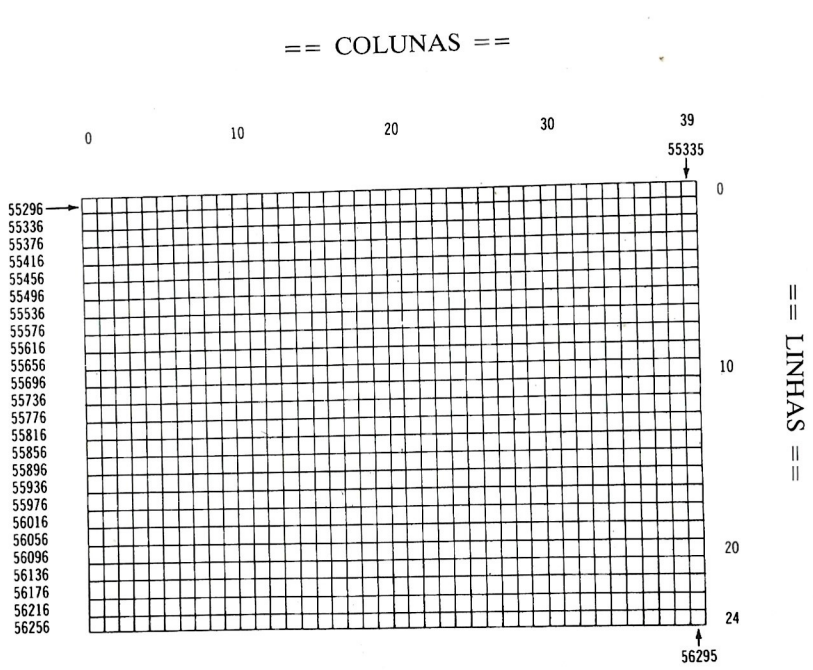
AND	IF A=B AND X=Y THEN STOP
OR	IF A=B OR X=Y THEN STOP
NOT	IF A=B NOT X=Y THEN STOP

Operadores Aritméticos

+	Sinal de adição
−	Sinal de subtracção
*	Sinal de multiplicação
/	Sinal de divisão
↑	Sinal de elevação
=	Sinal de igual
<	Menor que
>	Maior que
<=	Menor ou igual a
> =	Maior ou igual a
<>	Diferente de

Capítulo 5

Mapa de Memória do Visor e das Cores



POKES

Algumas localizações de memória para proteger os programas contra intrusos.

- Desactivar a tecla STOP
  POKE 808,251:POKE 808,239
- Anular a função LIST
  POKE 775,0:POKE 774,131:POKE 775,164
- Pôr o cursor a piscar
  POKE 204,0
- Desactivar o cursor
  POKE 204,1
- Cor fundo do écran, onde x é um número de 0 a 15
  POKE 53281,x
- Cor de bordo do écran, onde x é um número de 0 a 15
  POKE 53280,x
- Cor dos caracteres, onde x é um número de 0 a 15
  POKE 646,x
- Activar a tecla STOP
  POKE 808,237:POKE 808,237
- Activar a função LIST
  POKE 775,167
- Para impedir o uso do comando SAVE
  POKE 818,131:POKE 819,164
- Destruir o programa ao fazer SAVE ou LIST
  POKE 818,226:POKE 819,252
- O comando SAVE aparece no écran
  POKE 808,5
- Para provocar o reinício do programa, sempre que ocorra um erro.
  POKE 768,226:POKE 769,252
- Para repetir as teclas
  POKE 650,128
- Anular efeito de repetir as teclas
  POKE 650,127

Capítulo 6

Mensagens de "Erro"

Mensagens de Erro

Este capítulo trata das mensagens de erro enviadas para o écran, quando algo de anormal se passa na leitura do programa.

Para uma perfeita correcção de possíveis erros no programa, preste bastante atenção às mensagens.

BAD DATA Dados de string não foram reconhecidos numa abertura de file.

BAD SUBSCRIPT O valor especificado no comando DIM, está fora de valor.

CAN'T CONTINUE O comando CONT não trabalha porque o programa tem erro.

DEVICE NOT PRESENT Foi enviada uma instrução para uma saída que pode ser a impressora ou a disquete, sem estes estarem ligados.

DIVISION BY ZERO A divisão por zero não é permitida.

EXTRA IGNORED Os dois pontos foram introduzidos numa entrada input.

FILE NOT FOUND Não foi encontrado o programa solicitado.

FILE NOT OPEN A file especificada em CLOSE, CMD, PRINT # , INPUT # ou GET # , tem de ser aberta com o comando OPEN.

FILE OPEN Foi feita uma tentativa de abrir uma file que já se encontra aberta.

FORMULA TOO COMPLEX Uma expressão de string tem um valor muito grande, ou existe alguma fórmula que o computador não entende.

ILLEGAL DIRECT O comando INPUT só pode ser usado num programa e não em modo directo.

ILLEGAL QUANTITY Um número usado num argumento ou funções está fora de valor.

LOAD Há um problema com o programa do gravador.

NEXT WITHOUT FOR Este erro foi causado por uma má utilização dos ciclos FOR...NEXT

NOT INPUT FILE Foi feita uma tentativa com os comandos INPUT ou GET para enviar dados para a saída duma file, sem que esta tenha sido aberta.

NOT OUTPUT FILE Foi feita uma tentativa com o comando PRINT para enviar dados, sem que tenha sido dado o INPUT.

OUT OF DATA Um comando READ foi executado mas não havia dados no comando DATA.

OUT OF MEMORY Pode suceder este erro se o computador não tiver mais memória disponível, quando existirem muitos loops FOR... ou muitos GOSUBs... a trabalhar.

OVERFLOW O resultado duma fórmula é mais extenso que o devido.

REDIM'D ARRAY Um array só pode ser dimensionado uma vez, ou então quando o número de um elemento de array fôr superior a dez (10).

REDO FROM START Foram escritos caracteres numa entrada INPUT quando deviam ser números.

RETURN WITHOUT GOSUB Um comando RETURN foi encontrado sem a instrução GOSUB ter sido utilizada.

STRING TOO LONG Uma string só pode conter até 255 caracteres

?SINTAX ERROR Qualquer comando foi mal escrito.

TYPE MISMATCH Este erro ocorre quando um número é usado no lugar de uma string ou vice-versa.

UNDEF'D FUNCTION Funções mal definidas no programa.

UNDEF'D STATEMENT Foi executado um comando com as instruções RUN GOTO ou GOSUB sem existir o número de linha.

VERIFY O programa no gravador ou na disquete não é encontrado para ser feita a verificação.

Capítulo 7

Som e Música no "COMMODORE 64"

Introdução ao Sintetizador do *COMMODORE 64*

Este computador está equipado com o chip 6581 designado de SID (Sound Interface Device) que contém 3 vozes independentes e, é capaz de gerar efeitos de som compatíveis com o 6510 e outros microprocessadores da mesma família.

O SID tem uma alta resolução de controle da frequência, de volume e de harmónicos.

Especificações do chip SID:

3. . . . . Vozes independentes; valores de 0 a 4 KHZ
8. . . . . Oitavas
4. . . . . Formas de onda; triangular, serra, ruído e variável com impulso.
3. . . . . Moduladores de amplitude; valor 48 db
   - oscilador de sincronismo
   - controlo de som
   - filtros programáveis de altos e baixos.

A localização destes registos na memória do computador, estão entre o número 54272 a 54296 inclusivé.

A descrição das localizações é a seguinte:

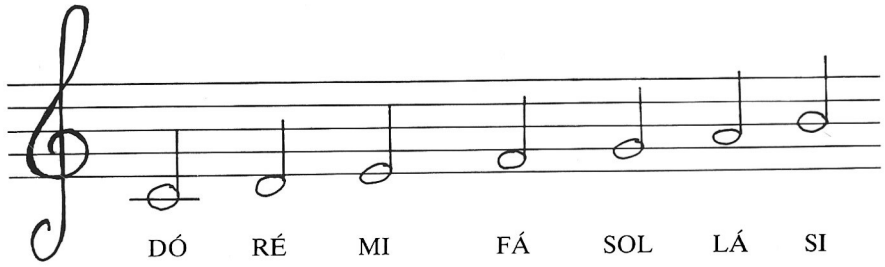
1.ª VOZ	2.ª VOZ	3.ª VOZ	EFEITOS PRODUZIDOS
54272	54279	54286	Valor de baixa frequência
54273	54280	54287	Valor de alta frequência
54274	54281	54288	Baixa vibração
54275	54282	54289	Alta vibração
54276	54283	54290	Controlo de registos
54277	54284	54291	Ataca o som na descaída
54278	54285	54292	Prolonga o nível do som

Para se criar efeitos de som, temos que utilizar valores de alta e baixa frequência, dentro das localizações de memória que pretendemos usar, recorrendo ao comando POKE.

Para criar simulação de vários instrumentos, seguem-se alguns exemplos:

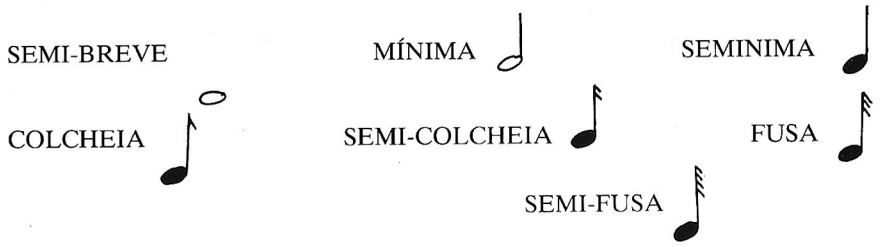
INSTRU-MENTOS	FORMAS DE ONDA	CONTROLE DESC. SOM	PROLONGA VOLUME	VALOR VIBRAÇÃO
PIANO	VARIÁVEL IMPUL.	9	0	A1-0 Ba-255
FLAUTA	TRIANG.	96	0	NÃO APL.
TROMPETE	DENTE DE SERRA	96	0	NÃO APL.
ACORDEÃO	TRIANG.	102	0	NÃO APL.
ÓRGÃO	TRIANG.	0	240	NÃO APL.
XILOFONE	TRIANG.	9	0	NÃO APL.

Os sons musicais são representados por sete notas que têm os seguintes nomes:



Os mesmos sinais que, segundo a sua composição na pauta, servem para indicar sons mais graves ou agudos, servem também para determinar conforme a figura que tomam, o tempo de duração da nota.

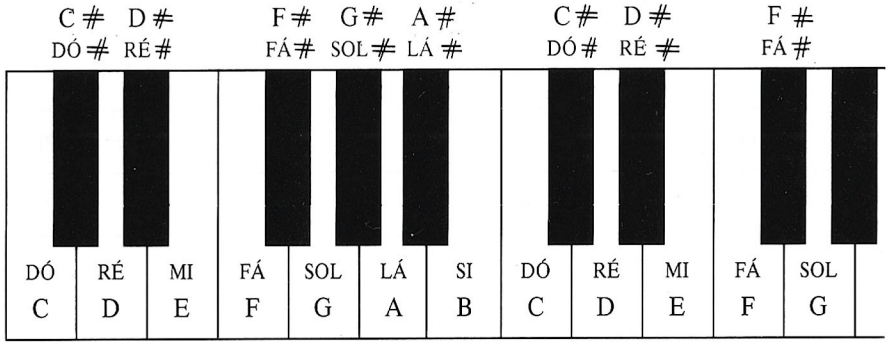
As figuras mais usadas de duração de som, são sete:



Cada figura vale o dobro da seguinte.

Todas estas notas podem ser ainda constituídas por meios tons. O intervalo entre o DÓ e o RÉ é de um tom. Por isso, para subir o DÓ meio tom, obteríamos o DÓ #.

ESCALA DO PIANO
COM AS NOTAS CORRESPONDENTES



O valor decimal para o tipo de onda das vozes, são os seguintes:

1 - VOZ POKE 54276
2 - VOZ POKE 54283
3 - VOZ POKE 54290

TRIANGULAR	17
DENTE DA SERRA	33
VARIÁVEL C/ IMPUL.	65
RUÍDO	129

VOLUME NO MÁXIMO:

POKE 54296, seguido de um número, compreendido de 0 (desligado) a 15 (máximo) de volume.

ATENÇÃO:

O volume tem de ser definido antes de gerar sons.

Valores das Notas Musicais

Vamos de seguida descrever a equivalência entre os números decimais e as notas musicais para as três vozes do computador em alta e baixa frequência.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
DO	DO#	RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI

NOTA	OITAVA	ALTA FREQU.	BAIXA FREQU.
0	C-0	1	18
1	C#-0	1	35
2	D-0	1	52
3	D#-0	1	70
4	E-0	1	90
5	F-0	1	110
6	F#-0	1	132
7	G-0	1	155
8	G#-0	1	179
9	A-0	1	205
10	A#-0	1	233
11	B-0	2	6
12	C-1	2	37
13	C#-1	2	69
14	D-1	2	104
15	D#-1	2	140
16	E-1	2	179
17	F-1	2	220
18	F#-1	3	8
19	G-1	3	54
20	G#-1	3	103
21	A-1	3	155
22	A#-1	3	210
23	B-1	4	12
24	C-2	4	73
25	C#-2	4	139
26	D-2	4	208
27	D#-2	5	25
28	E-2	5	103
29	F-2	5	185
30	F#-2	6	16
31	G-2	6	108
32	G#-2	6	206
33	A-2	7	53
34	A#-2	7	163
35	B-2	8	23
36	C-3	8	147
37	C#-3	9	21

NOTA	OITAVA	ALTA FREQU.	BAIXA FREQU.
38	D–3	9	159
39	D # –3	10	60
40	E–3	10	205
41	F–3	11	114
42	F # –3	12	32
43	G–3	12	216
44	G # –3	13	156
45	A–3	14	107
46	A # –3	15	70
47	B–3	16	47
48	C–4	17	37
49	C # –4	18	42
50	D–4	19	63
51	D # –4	20	100
52	E–4	21	154
53	F–4	22	227
54	F # –4	24	63
55	G–4	25	177
56	G # –4	27	56
57	A–4	28	214
58	A # –4	30	141
59	B–4	32	94
60	C–5	34	75
61	C # –5	36	85
62	D–5	38	126
63	D # –5	40	200
64	E–5	43	52
65	F–5	45	198
66	F # –5	48	127
67	G–5	51	97
68	G # –5	54	111
69	A–5	57	172
70	A # –5	61	126
71	B–5	64	188
72	C–6	68	149
73	C # –6	72	169
74	D–6	76	252
75	D # –6	81	161
76	E–6	86	105
77	F–6	91	140
78	F # –6	96	254
79	G–6	102	194
80	G # –6	108	223
81	A–6	115	88
82	A # –6	122	52
83	B–6	129	120
84	C–7	137	43
85	C # –7	145	83

NOTA	OITAVA	ALTA FREQU.	BAIXA FREQU.
86	D–7	153	247
87	D # –7	163	31
88	E–7	172	210
89	F–7	183	25
90	F# –7	193	252
91	G–7	205	133
92	G # –7	217	189
93	A–7	230	176
94	A # –7	244	103

ATENÇÃO:

No final do livro muitos programas foram desenvolvidos para melhor entendimento deste capítulo.

Capítulo 8

Introdução aos "SPRITES"

Introdução aos SPRITES

Em capítulos anteriores mostra-se como se usam símbolos gráficos do *COMMODORE 64* utilizando o comando PRINT para criar objectos e gráficos no écran. Falou-se também na utilização do comando POKE para colocar caracteres ou gráficos numa determinada posição do écran. Em ambos os casos, todos os objectos criados são limitados, uma vez só poderem usar os símbolos gráficos existentes no teclado do computador. Isto sem falarmos na dificuldade de querermos posicionar objectos no écran.

Tudo isto pode ser eliminado se usarmos nos jogos e programas, os gráficos definidos, por nós chamados SPRITES.

Um SPRITE é um objecto programável de alta resolução, podendo criar todo o tipo de figuras que se quer, utilizando os comandos BASIC.

Tudo o que temos de fazer é definir o objecto que queremos e determinar a sua localização no écran, tanto horizontal, como vertical e o computador fará o resto. Mas, isto não é tudo o que se pode fazer com os SPRITES; podemos mudar a cor, permitir a colisão com outro SPRITE, expandir o seu tamanho e colocá-los em movimento.

Agora o que temos de aprender sobre os SPRITES do *COMMODORE 64* é a sua localização e a melhor maneira de os utilizar. Contudo, é importante entender-se um pouco sobre o funcionamento interno do computador.

O computador guarda a informação através duma sequência de sinais eléctricos que se representam por 1s e 0s.

Cada célula de memória é representada por uma série de 8 (oito) 1s ou 0s, chamados dígitos binários ou bits. Podem também ser chamadas de bytes, quando se tem um conjunto de 8 (oito) bits.

Um bit, é uma fracção da informação que o computador pode guardar, quando o valor é 1 ou 0.

Quando a informação é dada ao computador, as teclas pressionadas são convertidas dentro de 8 bits em 1s (uns) e 0s (zeros) para a memória.

Como vamos trabalhar no sistema binário, onde os valores só podem ser 1s e/ou 0s, vamos fazer uma demonstração de como converter um número binário em decimal.

Quando todos os oito bits estão off, o conteúdo do BYTE é zero, quando todos estão on, o valor é 255. Atenção, pois a combinação dos números só podem ir de 0 a 255 inclusivé.

A conversão binário decimal é a seguinte:

	7	6	5	4	3	2	1	0
Posição binária:	2	2	2	2	2	2	2	2
Result. decimal:	128	64	32	16	8	4	2	1
Valores do bit.:	1	0	1	0	1	0	1	0

Agora temos que somar o resultado de cada bit que está ON para sabermos o resultado do BYTE:

128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	0

128 + 0 + 32 + 0 + 8 + 0 + 2 + 0 = 170

76543210 22222222	**Valor decimal**
00000000	0
00000001	1
00000010	2
00000100	4
00001000	8
00010000	16
00100000	32
01000000	64
10000000	128

Apresentamos de seguida um pequeno programa para fazer conversão de um BYTE para decimal

```
10 PRINT CHR$(147)
11 INPUT "entrada dos oito bits";A$
12 IF LEN(A$)     8 THEN PRINT "00000000 bits P.F.": GOTO 11
15 bi=0:c=0
20 FOR X=8 TO 1 STEP - 1:c=c + 1
30 bi=bi + val (MID$(A$,c,1))*2  (X-1)
40 NEXT X
50 PRINT A$;"em binário=";bi;"decimal"
60 GOTO 10
```

Os registos dos SPRITES são guardados num chip especial colocado dentro do *COMMODORE 64* e chamado de VIC II. Este chip contém uma série de bytes especiais, chamados REGISTOS, que foram criados especificamente para controlar os SPRITES.

Cada REGISTO tem uma função em separado, permitindo assim verificar se um SPRITE está activo ou inactivo, se está expandido ou normal.

A localização destes registos é a seguinte:

N.º REGISTO	DESCRIÇÃO
0–15	Posição do sprite
16	Movimento extra
21	Permite ao sprite estar na posição ON ou OFF
23	Expande o sprite (vertical)
27	Sprite prioritário
28	Multicor seleccionada
29	Expande o sprite (horizontal)
37–38	Multicores
39–46	Cores

Cada um destes registos, ou bytes, tem de ser nomeado na localização específica dentro da memória do computador para ser utilizado.

A localização de partida para o chip é POKE 53248,i.e., o endereço base do VIC II, atrás identificado, para que possamos criar gráficos, cores e movimentos dos SPRITES.

O chip VIC II, contém 46 registos e permite controlar até (8) oito SPRITES ao mesmo tempo.

Para desenhar, criar e mover os SPRITES, escolhemos um valor decimal correcto, colocando-se de seguida o comando POKE.

Se pretendemos o valor 4 no registo, faríamos:

POKE 53248 + 21,4

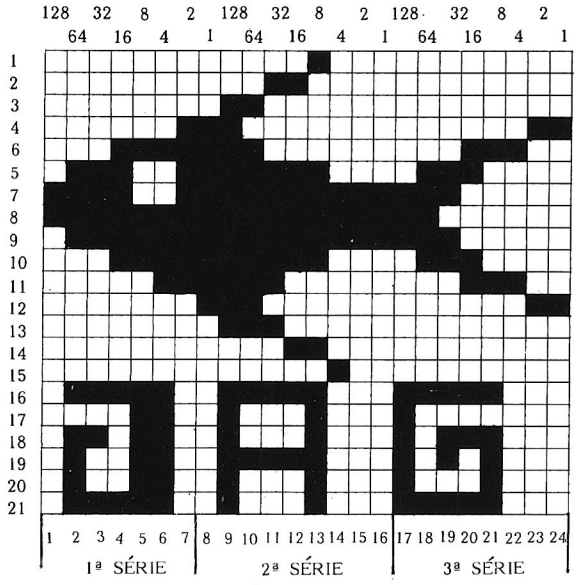
Desenhando SPRITES

Um SPRITE é feito de 504 pontos ou seja o resultado de um quadrado composto por 24 colunas × 21 linhas.

Como foi mencionado, pode-se utilizar os oito SPRITES simultaneamente. Cada ponto do quadrado corresponde a um bit; como temos 24 pontos temos uma linha de 24 bits, ou sejam, 3 bytes.

Para criar um SPRITE, temos de preencher os quadrados, correspondendo os quadros cheios ao valor de 1 e os vazios ao valor 0.

Note o desenho que se segue.



Depois de ter desenhado o gráfico pretendido, teremos de converter para decimal o resultado de cada série. O resultado da primeira linha, que é composto por três séries, seria:

128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1ª SÉRIE = 0								2ª SÉRIE = 8								3ª SÉRIE = ø							

Use este método para converter todas as séries de oito quadrados até à linha n.º 21. Uma vez calculados os valores, teremos que escrever um programa em BASIC para movimentarmos os SPRITES.

```
4 POKE 53280,0:POKE 53281,0
5 PRINT CHR$(147)"XXDEMONSTRAÇÃO DOS SPRITESXX"
6 FOR I=1 TO 22
7 PRINT TAB (13)"  ";TAB(23)"  "
8 NEXT I
10 V=53248:REM Activa chip VIC II
11 POKE 2042,13: POKE V+21,12
20 FOR J=832 TO 894:READ Q:POKE J,Q:NEXT J
30 FOR X=250 TO 1 SETP-1
32 IF X=126 THEN FOR P=1 TO 500:NEXT P
35 FOR I=1 TO 5
40 POKE V+4,150
50 POKE V+5,X
57 NEXT I
60 NEXT X
70 GOTO 30
100 DATA 0,8,0,0,48,0,0,192,0
110 DATA 3,128,3,31,198,28,115,248,112
120 DATA 243,255,224,255,255,192,127,255,224
130 DATA 31,248,112,7,224,28,1,192,3,0,224,0
140 DATA 0,24,0,0,4,0,124,248,248,12,136,192
150 DATA 108,136,184,76,248,184,76,136,136
160 DATA 124,136,248
```

Uma vez passado correctamente o programa, escreva o comando RUN, seguido de RETURN e verifique o efeito no écran.

Localização dos SPRITES

Os SPRITES são guardados na memória em bytes, tendo por isso que ser localizados com um número de referência.

A localização dos SPRITES é a seguinte

2040 para o SPRITE n.º 0
2041 para o SPRITE n.º 1
.
.
. e assim até...
.
.
2047 para o SPRITE n.º 7

Cada SPRITE pode conter um número até 255. Este número multiplicado por 64 corresponde ao endereço inicial dos dados do SPRITE.

Podemos colocar também uma das 16 cores disponíveis no * COMMODORE 64*, em cada SPRITE.

O mapa de registos do VIC II chip, contém um registo de cor para cada SPRITE, numerado de 39 a 46. O registo 39 marca a cor para o SPRITE 0, o registo 46 marca a cor do SPRITE 7.

Para mudar a cor do SPRITE n.º 2, escreva simplesmente POKE V+41,X, tendo X um valor de 0 a 15.

Agora sabemos desenhar um SPRITE, mudar a sua cor e localizações, resta-nos conhecer a forma de os situarmos onde quisermos no écran.

O mapa do registo do VIC II chip, tem as localizações 0 a 15 que são usadas para as coordenadas X,Y. O registo X posiciona o SPRITE na horizontal; o registo Y posiciona-o na vertical. Expandindo o SPRITE na vertical e horizontal, o registo 23 aumenta a sua largura e o registo 29 a sua altura.

Adicione estas linhas ao programa para expandir o SPRITE.

```
POKE V+23,12:POKE V+29,12
```

Demonstrando como se pode criar mais do que um SPRITE, acrescente as linhas que se seguem ao programa:

```
11 POKE V+21,12
12 POKE 2042,13:POKE 2043,13
45 POKE V+6,150
55 POKE V+7,255-X
```

NOTA:

Nas páginas finais, poderá encontrar mais programas.

Capítulo 9

Teclas c/ Função T + Cores do "COMMODORE 64"

Teclas com Funções

SHIFT + CLR/HOME	LIMPA ÉCRAN
CLR/HOME	COLOCA CURSOR NO TOPO ESQUERDO
SHIFT + crsr ↑	MOVE CURSOR PARA CIMA
SHIFT + crsr ←	MOVE CURSOR PARA A ESQUERDA
crsr ↓	MOVE CURSOR PARA BAIXO
crsr →	MOVE CURSOR PARA A DIREITA
CTRL + RVS/ON	IMAGEM EM MODO REVERSO
CTRL + RVS/OFF	FINALIZA MODO REVERSO
SHIFT + RUN/STOP	CARREGA UM PROGRAMA
RUN/STOP	PÁRA EXECUÇÃO DO PROGRAMA
SHIFT + COMMODORE	MUDA DE MAIÚSCULAS PARA MINÚSCULAS
SHIFT + INST/DEL	COLOCA UM CARÁCTER NA LINHA
INST/DEL	LIMPA UM CARÁCTER À ESQUERDA
SHIFT + GRÁFICO	IMPRIME GRÁFICOS DA DIREITA
COMMODORE + GRÁFICO	IMPRIME GRÁFICOS DA ESQUERDA
RUN/STOP + RESTORE	LIMPA TODOS OS DADOS DA MEMÓRIA

TECLAS PROGRAMÁVEIS

FUNÇÃO	VALOR DECIMAL	SIMBOLO
F1	CHR$(133)	
F2	CHR$(137)	
F3	CHR$(134)	
F4	CHR$(138)	
F5	CHR$(135)	
F6	CHR$(139)	
F7	CHR$(136)	
F8	CHR$(140)	

ATENÇÃO

No final deste livro estão incluídos programas de programação destas teclas.

Utilização das Cores do COMMODORE 64

O COMMODORE 64 tem a possibilidade de imprimir até 16 cores diferentes utilizando a combinação de algumas teclas.

TECLADO	COR	SIMBOLO
CTRL + 1	PRETO	
CTRL + 2	BRANCO	
CTRL + 3	VERMELHO	
CTRL + 4	CYAN	
CTRL + 5	PÚRPURA	
CTRL + 6	VERDE	
CTRL + 7	AZUL	
CTRL + 8	AMARELO	
COMMODORE + 1	LARANJA	
COMMODORE + 2	CASTANHO	
COMMODORE + 3	VERMELHO CLARO	
COMMODORE + 4	CINZENTO ESCURO	
COMMODORE + 5	CINZENTO MÉDIO	
COMMODORE + 6	VERDE CLARO	
COMMODORE + 7	AZUL CLARO	
COMMODORE + 8	CINZENTO CLARO	

ATENÇÃO

Todas as vezes que a tecla COMMODORE for mencionada corresponde ao simbolo C=

Abreviaturas dos Comandos BASIC

O BASIC do *COMMODORE 64* permite a introdução dos seus comandos através de abreviaturas, como por exemplo:

DIM = D + SHIFT I LOAD = L + SHIFT O RUN = R + SHIFT U

ABS	A	SHIFT	B	A	FOR	F	SHIFT	O	F
AND	A	SHIFT		A	FRE	F	SHIFT	R	F
ASC	A	SHIFT	S	A	GET	G	SHIFT	E	G
ATN	A	SHIFT	T	A	GOSUB	GO	SHIFT	S	GO
CHR$	C	SHIFT	H	C	GOTO		SHIFT	O	G
CLOSE	CL	SHIFT	O	CL	INPUT	I	SHIFT	N	I
CLR	C	SHIFT	L	C	LET	L	SHIFT	E	L
CMD	C	SHIFT	M	C	LEFT$	LE	SHIFT	F	LE
CONT	C	SHIFT	O	C	LIST	L	SHIFT	I	L
DATA	D	SHIFT	A	D	LOAD	L	SHIFT	O	L
DEF	D	SHIFT	E	D	MID$	M	SHIFT	I	M
DIM	D	SHIFT	I	D	NEXT	N	SHIFT	E	N
END	E	SHIFT	N	E	NOT	N	SHIFT	O	N
EXP	E	SHIFT	X	E	OPEN	O	SHIFT	P	O
PEEK	P	SHIFT	E	P	SPC(	S	SHIFT	P	S
POKE	P	SHIFT	O	P	SQR	S	SHIFT	Q	S
PRINT	?			?	STEP	ST	SHIFT	E	ST
PRINT	P	SHIFT	R	P	STOP	S	SHIFT	T	S
READ	R	SHIFT	E	R	STR$	ST	SHIFT	R	ST
RESTORE	RE	SHIFT	S	RE	SYS	S	SHIFT	Y	S
RETURN	RE	SHIFT	T	RE	TAB	T	SHIFT	A	T
RIGHT$	R	SHIFT	I	R	THEN	T	SHIFT	H	T
RND	R	SHIFT	N	R	USR	U	SHIFT	S	U
RUN	R	SHIFT		R	VAL	V	SHIFT	A	V
SAVE	S	SHIFT	A	S	VERIFY	V	SHIFT	E	V
SGN	S	SHIFT	G	S	WAIT	W	SHIFT	A	W
SIN	S	SHIFT	I	S					

Capítulo 10

Códigos ASCII e CHR$

Códigos ASCII e CHR$

Este mapa mostra os caracteres que se podem colocar no écran mediante o comando PRINT CHR(X), onde X é um valor decimal de 0 a 255.

Pode-se também obter esse valor decimal no écran com o comando PRINT ASC("X"), onde X é um carácter.

Exemplo:

Se pretendessemos escrever a palavra AMOR, utilizaríamos:

Letra A CHR$(65)
Letra M CHR$(77)
Letra O CHR$(79)
Letra R CHR$(82)

Para sabermos o valor decimal do número 1,2,3 faríamos o seguinte:

PRINT ASC("1");ASC("2");ASC("3");

O resultado seria:

49 para ASC("1")
50 para ASC("2")
51 para ASC("3")

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
	0	CRSR ↓	17	"	34	3	51
	1	RVS ON	18		35	4	52
	2	CLR HOME	19	$	36	5	53
	3	INST DEL	20	%	37	6	54
	4		21	&	38	7	55
WHT	5		22	.	39	8	56
	6		23	(	40	9	57
	7		24	)	41	:	58

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
DISABLES SHIFT C=	8		25	*	42	;	59
ENABLES SHIFT C=	9		26	+	43	<	60
	10		27	,	44	=	61
	11	RED	28	–	45	>	62
	12	CRSR →	29	.	46	?	63
RETURN	13	GRN	30	/	47	@	64
SWITCH TO LOWER CASE	14	BLU	31	0	48	A	65
	15	SPACE	32	1	49	B	66
	16	!	33	2	50	C	67
D	68	♠	97	π	126		155
E	69		98	◥	127	PUR	156
F	70		99		128	CRSR ↑	157
G	71		100		129	YEL	158
H	72		101		130	CYN	159
I	73		102		131	SPACE	160
J	74		103		132		161
K	75		104	f1	133		162
L	76		105	f3	134		163
M	77		106	f5	135		164
N	78		107	f7	136		165
O	79		108	f2	137		166
P	80		109	f4	138		167
Q	81		110	f6	139		168
R	82		111	f8	140		169
S	83		112	SHIFT RETURN	141		170
T	84	●	113	SWITCH TO UPPER CASE	142		171
U	85		114		143		172
V	86	♥	115	BLK	144		173
W	87		116	CRSR ↓	145		174
X	88		117	RVS OFF	146		175

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
Y	89		118	CLR HOME	147		176
Z	90		119	INST DEL	148		177
[	91		120		149		178
£	92		121		150		179
]	93		122		151		180
↑	94		123		152		181
←	95		124		153		182
	96		125		154		183
	184		186		188		190
	185		187		189		191

CÓDIGO 192 a 223 mesmo que 96 a 127
CÓDIGO 224 a 254 mesmo que 160 a 190
CÓDIGO 255 mesmo que 126

Programa de localização do valor decimal dos gráficos e caracteres do * COMMODORE 64*.

```
10 PRINT CHR$(147):REM limpa écran
20 FOR I=0 TO 255
30 PRINT " O CHR$("I") é igual a ";CHR$(I)
40 FOR Pausa=1 TO 500:NEXT Pausa
50 NEXT I
```

ATENÇÃO:

Para evitar que o écran seja limpo quando o CHR$(19) ou CHR$(147) fôr impresso, teremos que colocar uma linha no programa para isso não ser feito.

Depois de verificar o efeito que o programa produz, colocar a seguinte linha:

```
25 IF I=19 or I=147 THEN 40
```


Códigos do Écran

O mapa que se segue mostra todos os caracteres e gráficos que compõem o *COMMODORE 64*. Podemos imprimir estes caracteres e gráficos utilizando o comando POKE. A localização da memória para estes caracteres serem impressos é: 1024 a 2023

Neste mapa existem dois caracteres com o mesmo número decimal, mas só uma série de cada vez pode ser impressa no écran. A sua mudança pode ser efectuada se pressionarmos as teclas SHIFT e o símbolo COMMODORE simultaneamente.

Em BASIC podemos escrever POKE 53272,29 para maíusculas ou 1-série e POKE 53272,31 para minúsculas ou 2-série.

Pode-se também imprimir um carácter em modo reverso, bastando para o efeito adicionar 128 ao valor desejado.

Para obtermos um círculo na localização 1540, escrever-se-ía: POKE 1504,81.

Podemos mudar a cor do círculo utilizando o mapa de cor (55296-56295). Por exemplo, para mudar a cor do círculo para verde (código de cor 5) escrever POKE 55776,5.

```
10 FOR I=1 TO 20
20 POKE 1034+I,102
30 POKE 1134+I,102
40 POKE 1434+I,102
50 POKE 1714+I,102
60 NEXT I
```

1-SÉRIE	2-SÉRIE	POKE	1-SÉRIE	2-SÉRIE	POKE	1-SÉRIE	2-SÉRIE	POKE
@		0	C	c	3	F	f	6
A	a	1	D	d	4	G	g	7
B	b	2	E	e	5	H	h	8
I	i	9	%		37		A	65
J	j	10	&		38		B	66
K	k	11			39		C	67

1-SÉRIE	2-SÉRIE	POKE	1-SÉRIE	2-SÉRIE	POKE	1-SÉRIE	2-SÉRIE	POKE
L	l	12	(		40		D	68
M	m	13	)		41		E	69
N	n	14	*		42		F	70
O	o	15	+		43		G	71
P	p	16	,		44		H	72
Q	q	17	–		45		I	73
R	r	18	.		46		J	74
S	s	19	/		47		K	75
T	t	20	0		48		L	76
U	u	21	1		49		M	77
V	v	22	2		50		N	78
W	w	23	3		51		O	79
X	x	24	4		52		P	80
Y	y	25	5		53		Q	81
Z	z	26	6		54		R	82
[		27	7		55		S	83
£		28	8		56		T	84
]		29	9		57		U	85
↑		30	:		58		V	86
←		31	;		59		W	87
SPACE		32	<		60		X	88
!		33	=		61		Y	89
"		34	>		62		Z	90
#		35	?		63			91
$		36			64			92
		93			105			117
		94			106			118
		95			107			119
SPACE		96			108			120

1-SÉRIE	POKE	1-SÉRIE	POKE	1-SÉRIE	2-SÉRIE	POKE
	97		109			121
	98		110			122
	99		111			123
	100		112			124
	101		113			125
	102		114			126
	103		115			127
	104		116			

Os códigos de 128-255 são imagens em reverso dos códigos 0-127.

Capítulo 11

Programas

Programas

```
0 REM ******************************
1 REM *  DEMONSTRACAO DE SPRITES   *
2 REM *  E A SUA MOVIMENTACAO NO   *
3 REM *  ECRAN DE VARIAS FORMAS    *
5 REM ******-------------------******
6 REM      *     GRAFICOS      *
7 REM      *        DE         *
8 REM      *  J.A.GUIMARAES    *
10 REM     ********************
15 PRINT"■":POKE53280,12:POKE53281,0:GOSUB2000
17 FORJ=251*64TO(251*64+62):READA:POKEJ,A:NEXT
18 FORJ=253*64TO(253*64+62):READA:POKEJ,A:NEXT
19 FORJ=254*64TO(254*64+62):READA:POKEJ,A:NEXT
20 FORJ=252*64TO(252*64+62):READA:POKEJ,A:NEXT
25 FORJ=255*64TO(255*64+62):READA:POKEJ,A:NEXT
30 POKE 2040,251
32 POKE 2041,252
35 POKE 2042,254
36 POKE 2043,253
37 POKE 2044,255
40 A=208*256:I=0:POKE A+21,0
45 POKE A+29,31:POKE A+23,15
47 POKE A+21,31
50 FOR C=0 TO 3:POKE A+39+C,0:NEXT
60 POKE A+43,1
95 I=I+1:X=46
100 Y=X+I:F=F+1.5
105 POKE A+27,48
110 POKE A+1,169
115 POKE A+3,210
120 POKE A,IAND 255
122 POKE A+2,YAND 255
125 POKE A+4,IAND 255
150 POKE A+5,210
160 POKE A+6,YAND 255:POKE A+7,169
162 POKE A+8,FAND 255:POKE A+9,70
167 IFY>=256ANDI>=256THENPOKEA+16,15
168 IF Y>=256 AND I>=256 THEN 200
170 IF Y>=256 THEN POKE A+16,10:GOTO200
200 IF Y>=350 THEN POKE A+21,0:RUN
300 GOTO 95
500 DATA 0,0,0,0,0,0,0,0,8,0,0,15,0,0
503 DATA 11,0,0,8,0,2
505 DATA 58,0,1,62,0,0,157,0,7,72,0,14
507 DATA 184,0,29,208
510 DATA 0,59,248,0,55,212,0,111,186,0
512 DATA 239,125,0,222,240,1,221,224
520 DATA 1,189,192,3,251,192,2,27,128
530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,128
535 DATA 0,0,224,0,0
540 DATA 50,0,0,30,0,0,0,0,0,128,0,0
545 DATA 64,0,0,64,0,0
550 DATA 32,0,0,32,0,0,48,0,0,24,64,0
555 DATA 20,124,0,10
560 DATA 120,0,10,124,0,10,64,0,5,64,0
```



```
570 DATA 1,7,156,42,163,10,63,227,9,31
575 DATA 255,8,8,129
580 DATA 28,7,129,28,3,224,156,3,253
585 DATA 85,0,63,255,0,125,182,0
590 DATA 15,255,0,3,255,0,0,255,0,0
600 DATA 127,0,0,0,0,0,0,0,0,0,0,0,0,0
605 DATA 0,0,0,0,0,0,0,0
610 DATA 5,96,0,2,160,0,2,160,0,133,85
615 DATA 64,73,255,192
620 DATA 39,255,192,18,18,0,81,255,128
625 DATA 255,228,0,219,126,0,255,192,0
630 DATA 255,192,0,255,128,0,255,0,0
640 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
645 DATA 0,0,0,0,0,0,0,0,0,0,0,0
650 DATA 56,0,0,255,0,1,255,240,7,255
670 DATA 248,31,255,252,63,255,254,127
690 DATA 255,254,127,255,254,127,255
695 DATA 254,127,255,254,127
700 DATA 255,254,127,255,254,127,254
710 DATA 112,127,253,0,63,2,252,0,0,0
720 DATA 0,0,0,0,0,0,0,0,0,0,0,0
2000 PRINT"■■*  o  *    ●   .   *     *  *";
2002 PRINT". *  *  * . *    *   .  *  .  ";
2003 PRINT"■*  .  *     *  . *   .   *  .";
2005 PRINT"■   .  *     *   *  .    *  . ";
2006 PRINT"   *   *   .    *   o  *    ●";
2007 PRINT" ● .  *  o  .   *  .  *  ●   ";
2008 PRINT" * .  *  . *    *  .  "
2009 PRINT"■";TAB(11)"■◤■"
2010 PRINT"   ■◤■     ■◤  ■          ■◤■"
2020 PRINT"  ■◤  ■   ■◤    ■        ■◤  ■"
2030 PRINT" ■◤     ■ ■◤      ■      ■◤    ■"
2040 PRINT"■◤       ■          ■   ■◤      ■"
2050 PRINT"■                    ■ ■◤        ■";
2055 PRINTTAB(37)"■◤■
2060 PRINT"■                                ■";
2065 PRINTTAB(32)"■◤■  ■◤  ■";
2070 PRINT"■                                ■";
2075 PRINTTAB(30)"■◤  ■■◤    ";
2080 PRINT"■                                ";
2090 PRINT"■                                ";
2092 PRINT"■                                ";
2095 PRINT"■                                ";
2097 PRINTCHR$(5)
2100 FOR G=0 TO 119
2102 POKE 1664+G,160:POKE 55936+G,6
2110 POKE 1784+G,160:POKE 56056+G,6
2120 POKE 1904+G,160:POKE 56176+G,6
2140 NEXT
2500 RETURN:REM REGRESSA A LINHA 17
```

```
5 REM ********************
6 REM * SPRITE EM COLISAO *
7 REM ********************
20 FORM=251*64TO(251*64+62):READA:POKEM,A:NEXT
30 V=208*256
35 POKE 2040,13
```

```
40 POKE V+33,1:REM FUNDO BRANCO
45 POKE V+32,5:REM BORDO VERDE
50 POKE V+39,5
60 POKE V+23,0:REM NAO EXPANDE VERTICAL
65 POKE V+29,0:REM NAO EXPANDE HORIZONTAL
75 PRINT"▆"
90 PRINT"▆QQQQQQQQQQQ"
92 PRINT"╱╲"
95 PRINT"|  ╲___"
97 PRINT"╲        ╲_____"
98 PRINT" |               ╲"
99 PRINT" |                ╲_____"
100 PRINT" |                      ╲"
105 PRINT" |                       ╲"
110 PRINT" |                        ╲"
115 PRINT" |                         _____"
117 PRINT" |                              ╲"
118 PRINT" |                               ╲_____"
120 PRINT" |                                     ╲___"
122 PRINT" |________________________________________|"
123 PRINT"▆▆▆▆▆▆▆▆▆▆▆▆▆▆▆▆▆";TAB(29)"╲";
124 PRINTTAB(36)"╱"
125 FOR J=1 TO 8:PRINTTAB(29)" |";TAB(36)"| "
126 NEXT:PRINTTAB(29)" ______"
128 POKE V+21,1
130 X=20:Y=15
140 DX=2.6:DY=0:A=1.2
150 IFPEEK(V+31)=1ANDDY>0THENDY=-0.8*DY:Y=Y+DY
155 IF PEEK(V+31)=1ANDDY>0THENPOKEV+1,Y
160 X=X+DX:Y=Y+DY:DY=DY+A
170 POKE V+1,Y
180 POKE V+0,(XAND255)
190 IF X>=256 THEN POKE V+16,1:GOTO 210
200 POKE V+16,0
210 IF X<450 AND Y+DY<255 THEN 150
220 GOTO 130
500 DATA 0,0,0,0,0,0,0,112,0
510 DATA 3,254,0,15,255,128,31,255,192
520 DATA 63,255,224,63,255,224,127,255,240
530 DATA 127,255,240,127,255,240,63,255,224
540 DATA 63,255,224,31,255,192,15,255,128
550 DATA 3,254,0,0,112,0,0,0,0
560 DATA 0,0,0,0,0,0,0,0,0
```

```
1 REM **********************
2 REM *     * J.A.G. *     *
3 REM ******        ********
4 REM *BARRAS P/ SINTONIZAR*
5 REM *    A COR DO SEU    *
6 REM *      TELEVISOR     *
7 REM **********************
10 DIM C$(16)
15 C$(1)="▆R            ▆ ▆PRETO"
20 C$(2)="▆R            ▆ ▆BRANCO"
30 C$(3)="▆R            ▆ ▆VERMELHO"
40 C$(4)="▆R            ▆ ▆CYAN"
```



```
50 C$(5)="▆R            ▆ ▆PURPURA"
60 C$(6)="▆R            ▆ ▆VERDE"
70 C$(7)="▆R            ▆ ▆AZUL"
80 C$(8)="▆R            ▆ ▆AMARELO"
90 C$(9)="▆R            ▆ ▆LARANJA"
100 C$(10)="▆R            ▆ ▆CASTANHO"
110 C$(11)="▆R            ▆ ▆VERMELHO CLARO"
120 C$(12)="▆R            ▆ ▆CINZENTO ESCURO"
130 C$(13)="▆R            ▆ ▆CINZENTO MEDIO"
140 C$(14)="▆R            ▆ ▆VERDE CLARO"
150 C$(15)="▆R            ▆ ▆AZUL CLARO"
160 C$(16)="▆R            ▆ ▆CINZENTO CLARO"
200 PRINT"▆";:BO=53281:POKEBO-1,5:POKEBO,12
210 PRINTTAB(8)"▆BARRAS DE COR▆▆▆"
220 FOR I=1 TO 16
230 PRINTTAB(8)C$(I)
250 GOSUB 500
300 NEXT I
310 FOR U=2 TO 15
320 POKE BO,U
330 GOSUB 500
340 NEXT U
350 GOTO 350
500 FOR PAUSA=1 TO 500:NEXT PAUSA
510 RETURN

READY.


READY.
```

```
10 REM ****************
11 REM *  LEITURA DO  *
20 REM * CODIGO ASCII *
30 REM *  DO TECLADO  *
40 REM *  COM ROTINA  *
50 REM *  DE MUSICA   *
60 REM ****************
70 PRINTCHR$(8)
100 PRINT"▆            PRESSIONA UMA TECLA"
500 GETA$:IF A$="" THEN 500
550 X=ASC(A$)
600 IF X=147ORX=19ORX=13 OR X=148THEN 500
602 IF X=20 THEN 500
605 PRINT"▆QQQQQQQ"
607 PRINTTAB(8);"╭──────────────────────────╮"
608 PRINTTAB(8);"|   ▆ CODIGO ASCII ▆        |"
609 PRINTTAB(8);"|        ▆DE▆               |"
610 PRINTTAB(8);"|                           |"
611 PRINTTAB(8);"|                           |"
612 PRINTTAB(8);"|     =              =      |"
613 PRINTTAB(8);"|                           |"
614 PRINTTAB(8);"|                           |"
615 PRINTTAB(8);"╰──────────────────────────╯"
625 PRINTTAB(10);"▆▆▆▆▆CHR$("X;")";
627 PRINTTAB(22);"ASC('"A$;"')"
630 PRINT:PRINTTAB(14);A$;TAB(25);X
650 GOSUB 1000:GOTO 500
```

```
1000 V=54272
1010 POKE V+4,0:POKE V+11,33:POKE V+8,0
1020 POKEV+24,15:POKEV+19,X:POKEV+20,X
1025 POKEV+17,X:POKEV+18,65
1030 POKE V+5,10:POKE V+1,X:POKE V+4,17
1040 POKE V+12,64
1060 FOR G=1 TO 150:NEXT
1100 RETURN

READY.


10 REM ********************
15 REM *     CONVERSAO      *
20 REM * DECIMAL/BINARIO *
25 REM ********************
30 POKE 53280,0:POKE 53281,0:Y=3
40 PRINTCHR$(5);"■■■ CONVERSAO ■■"
45 PRINTTAB(11)"■ DECIMAL/BINARIO ■"
47 PRINTTAB(28)"■■■■ POR J.A.G. ■■"
50 A$=CHR$(17):B$="■                    ■"
60 C$=CHR$(157)
70 PRINT"■■■■   INTRODUZA UM No DE (0-255)";
80 INPUT N$
85 N=VAL(N$)
90 IFN<0 OR N>255 THEN 300
130 PRINTTAB(Y)A$;A$;B$
140 FOR I=7 TO 0 STEP-1
150 R=INT(2↑I):
170 C=INT(N/R)
180 N=N-C*R
190 PRINTTAB(Y)"■";C;C$;
250 NEXT I:PRINT
260 A$=A$+CHR$(17):Z=Z+1
270 IF Z=8 THEN Y=20:GOTO 50
280 IF Z=16 THEN 400
300 PRINT"■"
310 PRINTTAB(27)"■          "
320 GOTO 70
400 PRINT"■      ■CONVERSAO COMPLETA ";
405 PRINT"MAIS (S/N)■"
410 GET B$:IF B$="" THEN 410
420 IF B$="S" THEN RUN
430 IFB$="N"THENPRINT"* FIM *";:GOTO430
440 GOTO 410

READY.


1 PRINT"■"
5 REM ************
6 REM * IMITANDO *
7 REM *   UMA    *
8 REM * SIRENE   *
9 REM ************
10 SD=54272
20 POKE SD+24,15
25 POKE SD+6,128
30 POKE SD+5,0
35 POKE SD+12,129
```



```
40 POKE SD+4,17:POKE SD+16,33
50 FOR X=8 TO 15
60 FOR N=50-C TO 30-C+1 STEP-1
70 POKE SD+1,N
75 POKE SD+10,X
80 NEXT N
90 NEXT X
100 GOTO 50

 EADY.

0 REM ***********************************
1 REM * ROTINAS DE SOM P/DEMONSTRACAO *
2 REM *  DAS CAPACIDADES DO SID CHIP   *
3 REM * COMO SINTETIZADOR ELECTRONICO *
4 REM ***********************************
5 POKE 53281,11:POKE 53280,11:POKE646,0
6 S=54272:S1=54279:S2=54286:WV=33
7 C=0:PRINT"■"TAB(12)"EFEITOS DE SOM"
8 PRINT"■■■■■        1 -- TOCA UM SOM"
10 PRINT"■        2 -- TOCA TODOS OS SONS
12 PRINT"■        3 -- MUDA FORMA DE ONDA
14 PRINT"■        4 -- FIM"
16 POKE 198,0
18 GET A$:IF A$="" THEN 18
20 IF A$="1" THEN 30
22 IF A$="2" THEN C=1:GOTO 60
24 IF A$="3" THEN 36
26 IF A$="4" THEN PRINT"■":END
28 GOTO 18
30 INPUT"■ENTRADA DO SOM(1-7)";N
32 IF N<1 OR N>7 THEN 30
34 ON N GOTO 62,86,110,136,162,192,218
36 PRINT"■    FORMAS DE ONDA"
38 PRINT"■■1 -- TRIANGULAR"
40 PRINT"■2 -- DENTE DE SERRA"
42 PRINT"■3 -- COM OSCILACAO"
44 PRINT"■4 -- COM RUIDO"          6
46 GET A$:IF A$="" THEN 46
48 IF A$="1" THEN WV=17:GOTO 6
50 IF A$="2" THEN WV=33:GOTO 6
52 IF A$="3" THEN WV=65:GOTO 6
54 IF A$="4" THEN WV=129:GOTO 6
56 GOTO 46
58 :
60 REM
62 GOSUB 244
64 POKE S+5,88:POKE S+6,88
68 POKE S+4,WV
70 FOR I=0 TO 200 STEP 0.5
72 POKE S+1,I:POKE S+1,I+55
74 NEXT I
76 POKE S+4,WV-1
78 IF C=1 THEN 86
80 GOTO 6
82 :
84 REM 2 SOM
86 GOSUB 244
88 POKE S+1,130
```

```
90 POKE S+5,204
92 POKE S+15,30
96 FOR I=1 TO 300:POKE S+4,21
98 POKE S+4,WV
100 NEXT I
102 IF C=1 THEN 110
104 GOTO 6
106 :
108 REM 3 SOM
110 GOSUB 244
112 POKE S+5,9:POKE S+6,5
114 FOR I=1 TO 200
116 HF=INT(RND(1)*100)
118 LF=INT(RND(1)*100)
120 POKE S+1,HF:POKE S+1,LF
122 POKE S+4,WV
124 POKE S+4,WV-1
126 NEXT I
128 IF C=1 THEN 136
130 GOTO 6
132 :
134 REM 4 SOM
136 GOSUB 244
138 POKE S+5,8:POKE S+6,204
140 POKE S1+5,8:POKE S1+6,204
142 FOR I=1 TO 100
144 POKE S+1,I:POKE S1+1,I
146 POKE S+4,WV:POKE S1+4,129
148 FOR J=1 TO 5:NEXT J
150 POKE S+4,WV-1:POKE S1+4,128
152 NEXT I
154 IF C=1 THEN 162
156 GOTO 6
158 :
160 REM 5 SOM
162 GOSUB 244
164 POKE S+5,255:POKE S+6,204
166 POKE S1+5,64:POKE S1+6,234
168 POKE S2+5,0:POKE S2+6,204
170 POKE S+1,2:POKE S,24
172 POKE S1+1,8:POKE S,97
174 POKE S2+1,67:POKE S2,15
176 POKE S+4,WV
178 FOR I=1 TO 3000:NEXT:POKE S1+4,WV
180 FOR I=1 TO 2000:NEXT:POKE S2+4,WV
182 FOR I=1 TO 2500:NEXT:POKE S2+4,WV-1
183 POKE S1+4,WV-1:POKE S+4,WV-1
184 IF C=1 THEN 192
186 GOTO 6
188 :
190 REM 6 SOM
192 GOSUB 244
194 POKE S+5,9:POKE S+6,0
196 POKE S1+5,9:POKE S1+6,0
198 FOR I=1 TO 150
200 POKE S+1,I:POKE S1+1,I+1
202 POKE S1+1,151-I:POKE S1+1,152-I
204 POKE S+4,WV:POKE S1+4,WV
206 POKE S+4,WV-1:POKE S1+4,WV-1
208 NEXT I
210 IF C=1 THEN 218
212 GOTO 6
```



```
214 :
216 REM 7 SOM
218 GOSUB 244
222 POKE S+5,88:POKE S+6,204
224 POKE S+4,WV
226 FOR J=1 TO 15
228 FOR I=40 TO 0 STEP-1
230 POKE S+1,I
232 NEXT I
234 POKE S+4,WV-1
236 NEXT J
238 GOTO 6
240 :
242 REM LIMPA O SID CHIP
244 FOR I=0 TO 24:POKE S+I,0:NEXTI
246 POKE S+24,15:POKE S+3,8:POKES+2,0
250 RETURN

READY.

10 REM ********************
15 REM * MUDANCA DA COR *
20 REM *  DE TODOS OS   *
30 REM * CARACTERES NO  *
40 REM *      ECRAN      *
50 REM ********************
100 FOR I=50050 TO 50083:READ Z
102 POKE I,Z:C=C+Z:NEXT
105 IF C<> 6247 THEN PRINT"ERRO"
107 IF C<> 6247 THEN PRINT"VERIFIQUE"
108 IF C<> 6247 THEN LIST 119-
110 PRINT"■      OPERACAO DO PROGRAMA"
115 PRINT"■■     ESCREVE O COMANDO     "
116 PRINT"        SYS 50050,X
117 PRINT"  ONDE X E O CODIGO DA COR"
119 DATA 32,155,183,165,101,162
120 DATA 0,134,252,162,216,134,253
140 DATA 145,252,230,252,166,252,208
150 DATA 2,230,253,224,232,208,242
160 DATA 166,253,224,219,208,236,96



1 REM ****************************
2 REM * PROGRAMA P/ DEMONSTRACAO *
3 REM * DA ROTINA DE INTRODUCAO  *
4 REM *           ALFA           *
5 REM ****************************
10 DIM X(100)
15 PRINT"■       ENTRADA DE FICHEIROS "
20 PRINT"■NOME:":INI=5:CO=30:GOSUB 2000
25 NO$(X)=N$:PRINT:PRINT
30 PRINT"MORADA:":INI=7:CO=30:GOSUB 2000
35 MO$(X)=N$:PRINT:PRINT
40 PRINT"No TELF:":INI=9:CO=28:GOSUB 2000
45 TE$(X)=N$:PRINT:PRINT
50 PRINT"C.POSTAL:":INI=10:CO=28:GOSUB 2000
55 CO$(X)=N$:PRINT:PRINT:X=X+1
500 PRINTTAB(9)"■■■■■NOVA ENTRADA (S/N)"
510 GET A$:IF A$="" THEN 510
520 IF A$="S" THEN 15
530 IF A$="N" THEN 600
540 GOTO 510
```

```
600 PRINT"■        FICHEIRO NO ECRAN"
620 FOR I=0 TO X
630 PRINTTAB(2)NO$(I)
640 PRINTTAB(2)MO$(I)
650 PRINTTAB(2)TE$(I)
800 NEXT
900 END
1500 REM *****************************
1510 REM * ROTINA DE INTRODUCAO ALFA *
1512 REM **********   POR   ***********
1515 REM *   JOSE ARAUJO GUIMARAES   *
1517 REM *****************************
1520 REM *  ESTA ROTINA PERMITE DAR   *
1530 REM *  ENTRADA  DE VALORES SEM   *
1540 REM *  UTILIZARMOS O COMANDO     *
1550 REM *         == INPUT ==        *
1560 REM *****************************
2000 PRINT"■":REM CURSOR P/CIMA
2010 A$="":N$=""
2015 FORI=1TO CO:PRINTTAB(INI)".";:NEXT
2020 PRINT"■"
2030 FOR I=1 TO CO
2040 GET A$:IF A$="" THEN 2040
2050 IF A$=CHR$(13) THEN RETURN
2060 IF A$=CHR$(157) THEN 2100
2065 IFA$=CHR$(29)OR A$=CHR$(20)THEN2100
2080 IF ASC(A$)<31ORASC(A$)>95 THEN2040
2090 PRINTTAB(INI)A$;:N$=N$+A$:NEXT:RETURN
2100 IF I<2 THEN 2040
2110 PRINTCHR$(157)"."CHR$(157);:I=I-1
2120 N$=LEFT$(N$,I-1):GOTO 2040

READY.

5 REM ***************************
6 REM * PROGRAMA P/DEMONSTRACAO *
7 REM * DE ORDENACAO ALFABETICA *
8 REM *    ATE 100 PALAVRAS     *
9 REM ***************************
10 DIM A$(100)
15 POKE 53281,15:POKE 53280,12:N=1
20 PRINT"■■";TAB(14)"■ ORDENADOR ■"
25 PRINTTAB(9)"■-- A FRASE P/ORDENAR --"
27 PRINTTAB(14)"OU ■ F1 ■ P/FIM ■■■■"
28 PRINT TAB(14)N;" - FRASE -"
29 FOR Y=1 TO 39:PRINT".";:NEXT
31 PRINT"■":X$="":Z$=""
32 FOR Y=1 TO 39
33 GET X$:IF X$="" THEN 33
34 IF X$="■"THEN X$="":N=N-1:GOTO 55
35 IF X$=CHR$(13) THEN 50
37 IF X$=CHR$(29)OR X$=CHR$(157)THEN 20
39 IF ASC(X$)<31 OR ASC(X$)>95 THEN 33
40 PRINTX$;:Z$=Z$+X$
45 NEXT Y
50 A$(N)=Z$:N=N+1
52 IFN>=100THEN PRINT"■■ORDENACAO MAXIMA"
53 IF N>=100 THEN END
54 GOTO 20
55 E=0
60 FOR I=1 TO N-1
70 IF A$(I)<=A$(I+1)THEN 120
80 C$=A$(I)
```



```
90 A$(I)=A$(I+1)
100 A$(I+1)=C$
110 E=1
120 NEXT I
130 IF E=1 THEN 55
135 PRINT"■";TAB(10)"LISTAGEM ORDENADA"
140 X=0
150 FOR I=1 TO N
170 PRINT"■"A$(I)
175 X=X+1
180 IF X>=22 THEN X=0:GOSUB 300
190 NEXT I
200 GOTO 200
300 PRINT:PRINT"  PRESSIONE UMA TECLA";
305 PRINT" PARA NOVO ECRAN "
310 GET C$:IF C$="" THEN 310
320 PRINT"■"
330 FOR T=1 TO 23
340 PRINT"
350 NEXT:PRINT"■"
400 RETURN

READY.

1 REM PROGRAMA ESCRITO PARA TRABALHAR
2 REM COM UMA IMPRESSORA EPSON GX-80
3 REM COMO M/DE ESCREVER PODENDO-SE
4 REM ESCREVER EM REVERSO MODO OU
5 REM COM OS CARACTERES AMPLIADOS:
6 REM PROGRAMA IDEALIZADO E ESCRITO P/
7 REM JOSE ARAUJO GUIMARAES
10 RV$=CHR$(18):RO$=CHR$(146):A2$=RO$
15 DB$=CHR$(14):OF$=CHR$(15):A1$=OF$
20 POKE53281,1:POKE53280,8:CR$=CHR$(157)
25 OPEN 3,4,7
30 DIM X$(100)
35 N=0
40 PRINTCHR$(14);CHR$(8):GOSUB 3000
60 PRINT"■■‾‾♥Γ/ |♠-80 \/‾SCREVER DE";
70 PRINT"  Ύ‾♥‾ | ⌒\♠—♠‾♥■";
100 PRINT"■■■■■■■■■■■■■■■■■■■■■■
120 PRINT"■ ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
125 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
130 PRINT" ▒—3 \ODO REVERSO ▒ —5 ♠UMEN";
135 PRINT"TA LARGURA▒"
140 PRINT" ▒—4 /ORMAL          ▒ —6 /ORMAL";
145 PRINT"              ▒"
150 PRINT" ▒▒▒▒ ■7/‾ORRIGIR \ARGENS | E";
155 PRINT"CLE (—7) ■▒▒▒▒▒▒■";
160 PRINT"■"
200 REM INICIO DO PROGRAMA
400 INI=ME:CO=MD:GOSUB 1500
404 N2=INT(ME/256):N1=ME*6-(N2*256)
420 IFN$=""THENPRINT#3:GOTO200
500 PRINT#3,CHR$(27)CHR$(16)CHR$(N2);
505 PRINT#3,CHR$(N1)A2$A1$N$
800 N=N+1
900 IF N>8 THEN PRINT"■■":N=0:GOSUB 4000
905 IF N>8 THEN 160
1000 PRINT:GOTO 200
1500 REM *****************************
1510 REM * ROTINA DE INTRODUCAO ALFA *
1512 REM **********   POR   ***********
```

```
1515 REM *   JOSE ARAUJO GUIMARAES   *
1520 REM ******************************
2000 PRINT"▯":REM CURSOR P/CIMA
2010 A$="":N$=""
2015 FORI=1 TO CO:PRINTTAB(INI)".";:NEXT
2020 PRINT"▯▯":REM CURSOR P/CIMA
2030 FOR I=1 TO CO
2040 GET A$:IF A$="" THEN 2040
2045 IF I>CO-10 THEN GOSUB 5000
2050 IF A$=CHR$(13) THEN RETURN
2055 IF A$=CHR$(147) THEN 2040
2060 IF A$=CHR$(157) THEN 2100
2063 IF A$=CHR$(148) THEN 2100
2065 IF A$=CHR$(29) THEN2100
2070 IF A$=CHR$(145) THEN 2040
2073 IFA$=CHR$(134)THEN A2$=RV$:GOTO2040
2075 IFA$=CHR$(135)THENA1$=DB$:GOTO2040
2077 IFA$=CHR$(138)THENA2$=RO$:GOTO2040
2080 IFA$=CHR$(139)THENA1$=OF$:GOTO2040
2082 IFA1$=DB$ THEN PRINT" ";
2085 IFA$=CHR$(136)THEN RUN
2090 IF ASC(A$)<31 THEN 2040
2095 PRINTTAB(INI)A$;:N$=N$+A$:NEXT:RETURN
2100 IF I<2 THEN 2040
2110 PRINTCHR$(157)"."CHR$(157);:I=I-1
2120 N$=LEFT$(N$,I-1):GOTO 2040
2200 RETURN
3000 PRINT"▯";
3002 PRINTTAB(14)"▯ ⌐NSTRUCOES ▯"
3010 ME=2:MD=74:PRINT
3012 PRINT"▯▯  ----------------------▯▯-┌┐└ ▯ DE";
3013 PRINT" ┐▯┌┐└▯▯------------------   ▯";
3015 FOR Y=1 TO 16
3020 PRINT"▯  -------------------------";
3022 PRINT"----------------------------   ▯";
3025 NEXT Y
3027 PRINT"¯STAS SAO AS MARGENS PROGRA";
3030 PRINT"MADAS:¬ARA ALTERA-LAS PRES";
3032 PRINT"SIONAR A TECLA CORRESPONDEN";
3035 PRINT"TE MOVENDO UM ESPACO DE CADA";
3037 PRINT" VEZ"
3040 FOR T=0 TO 39:PRINT"=";:NEXTT
3070 PRINT" -1 ⌐ARGEM ¯SQUERDA | ";
3080 PRINTTAB(23)"-2 ⌐ARGEM ¯IREITA"
3085 PRINTTAB(10)"|ARRA ¯SPACOS |ERMINA";
3100 PRINT"▯▯▯▯▯▯▯▯▯▯"
3110 FOR V=2 TO 36
3115 GET G$:IF G$="" THEN 3115
3120 IFG$=CHR$(133)THENME=ME+1:GOTO3200
3130 IFG$=CHR$(137)THEN MD=MD-1:ES=ES+1
3135 IF G$=" " THEN 4000
3140 PRINTTAB(39-ES)CR$"▯  "CR$;:GOTO3300
3200 PRINTTAB(ME-2)"▯  |";:GOTO 3300
3300 PRINT"▯▯":REM CURSOR P/CIMA
3305 PRINT;
3310 NEXT
4000 PRINT"▯▯":REM HOME+CURSOR P/BAIXO
4010 FOR T1=1 TO 18
4020 PRINT"
```



```
4030 NEXT
4050 RETURN
5000 REM ROTINADESOMP/INDICARFIMDE M.D
5005 S=212*256
5010 POKE S+4,0:POKE S+18,0:POKE S+11,0
5020 POKE S+24,15:POKE S+19,0
5030 POKE S+20,148:POKE S+17,88
5040 POKE S+18,5:POKE S+5,10
5050 POKE S+1,14:POKE S+4,33
5060 RETURN

READY.
```